

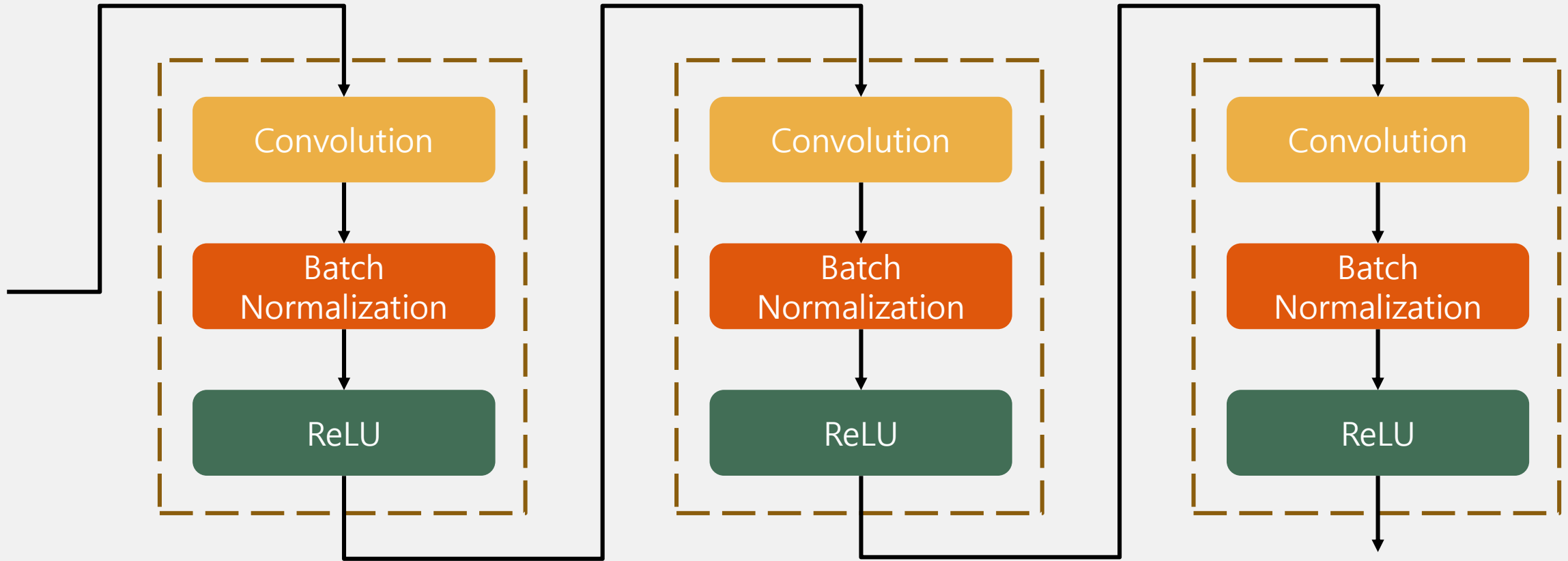


# Famous CNN Models



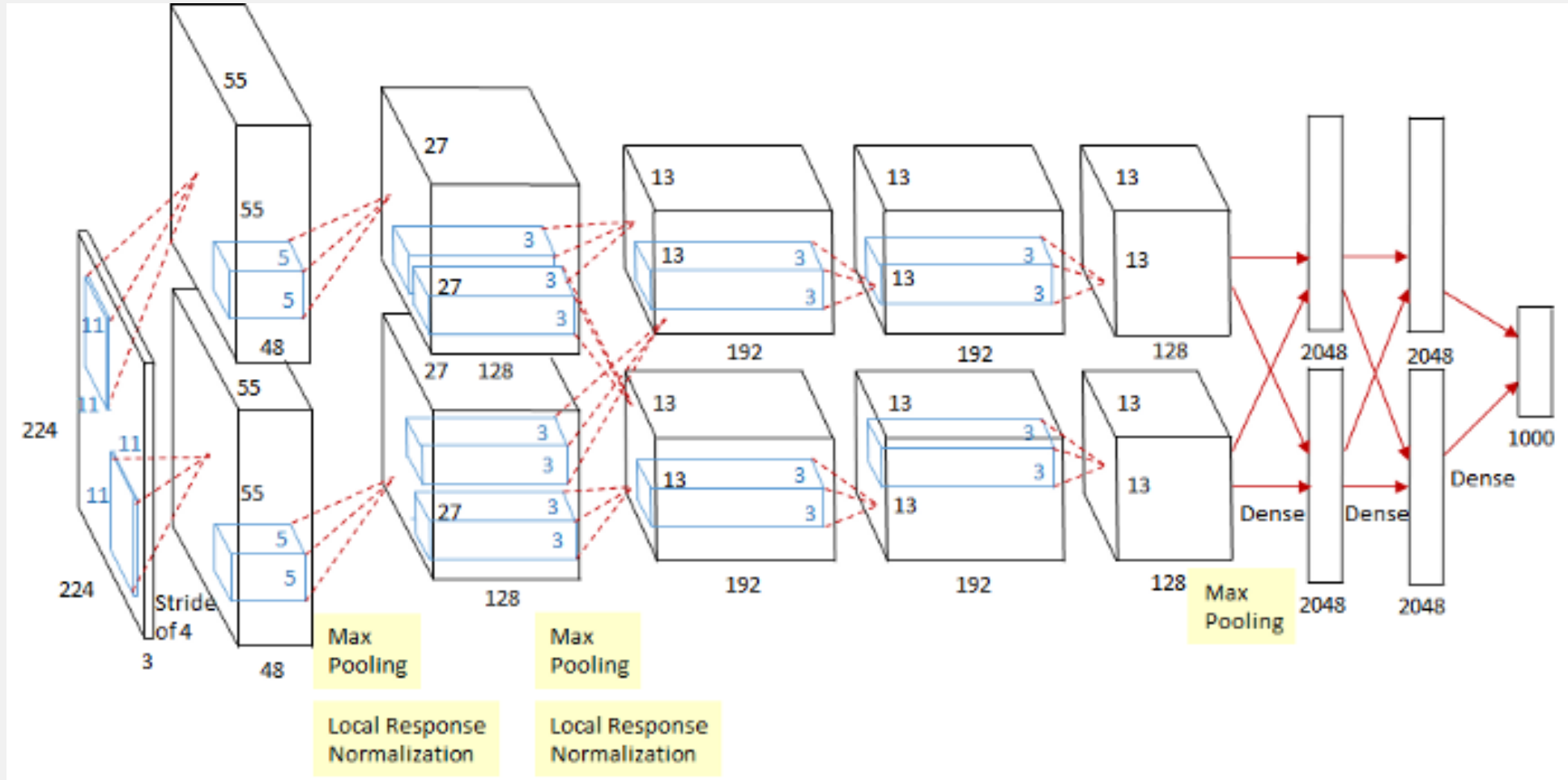
# CNN Model

A general block and architecture of CNN



[Reference](#)

# CNN Model

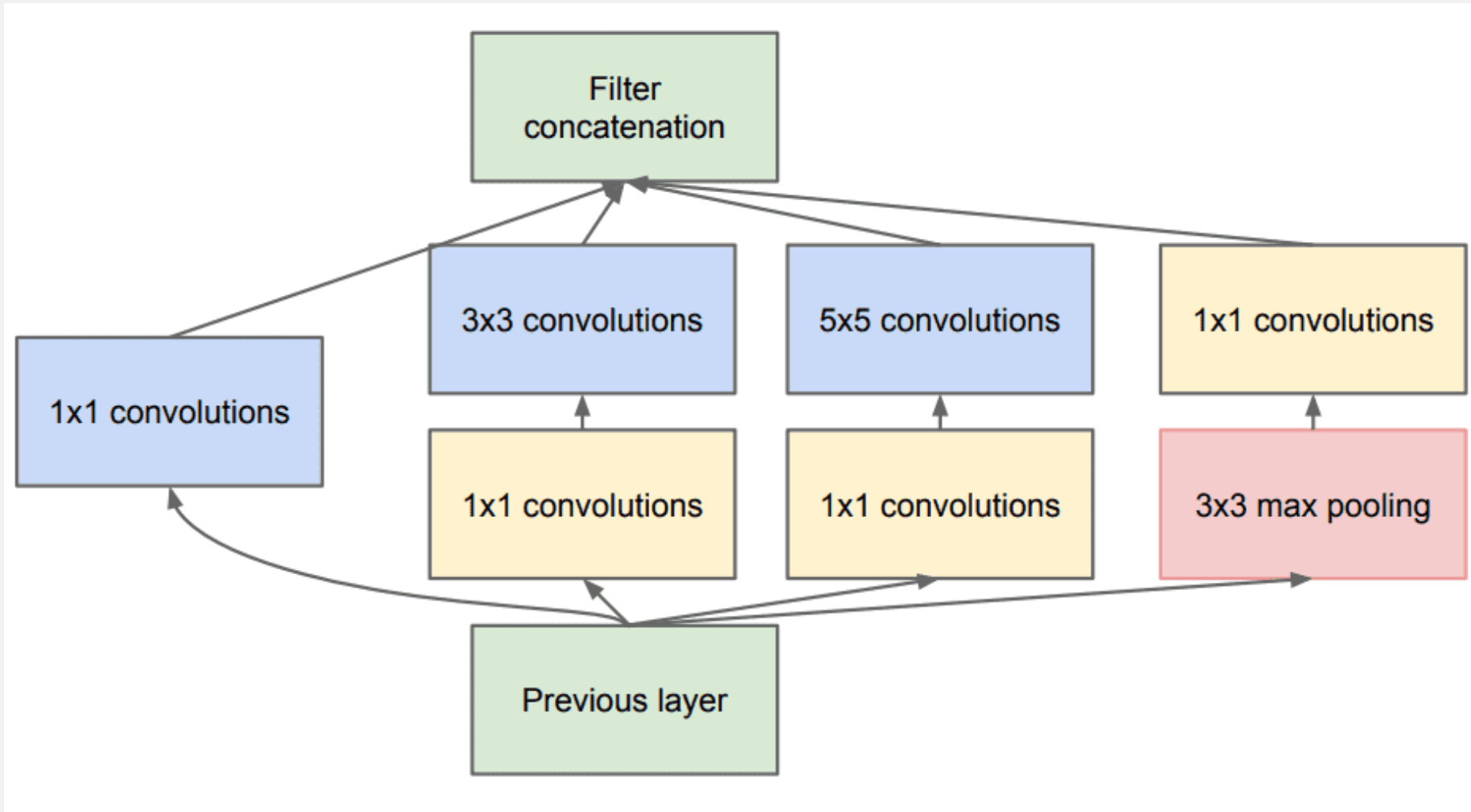


[AlexNet](#)

# CNN Model

										<u>Image</u>														
														Number of Parameters (millions)	Top-5 Error Rate (%)									
Image	Conv3-64	Max pool	Conv3-128	Max pool	Conv3-256	Conv3-256	Max pool	Conv3-512	Conv3-512	Max pool	Conv3-512	Conv3-512	Max pool	FC-4096	FC-4096	FC-1000	Soft-max	133	10.4					
<b>VGG-11</b>																								
Image	Conv3-64	LRN	Max pool	Conv3-128	Max pool	Conv3-256	Conv3-256	Max pool	Conv3-512	Conv3-512	Max pool	Conv3-512	Conv3-512	Max pool	FC-4096	FC-4096	FC-1000	Soft-max	133	10.5				
<b>VGG-11 (LRN)</b>																								
Image	Conv3-64	Conv3-64	Max pool	Conv3-128	Conv3-128	Max pool	Conv3-256	Conv3-256	Max pool	Conv3-512	Conv3-512	Max pool	Conv3-512	Conv3-512	Max pool	FC-4096	FC-4096	FC-1000	Soft-max	133	9.9			
<b>VGG-13</b>																								
Image	Conv3-64	Conv3-64	Max pool	Conv3-128	Conv3-128	Max pool	Conv3-256	Conv3-256	<b>Conv1-256</b>	Max pool	Conv3-512	Conv3-512	<b>Conv1-512</b>	Max pool	Conv3-512	Conv3-512	<b>Conv1-512</b>	Max pool	FC-4096	FC-4096	FC-1000	Soft-max	134	9.4
<b>VGG-16 (Conv1)</b>																								
Image	Conv3-64	Conv3-64	Max pool	Conv3-128	Conv3-128	Max pool	Conv3-256	Conv3-256	Conv3-256	Max pool	Conv3-512	Conv3-512	Conv3-512	Max pool	Conv3-512	Conv3-512	Conv3-512	Max pool	FC-4096	FC-4096	FC-1000	Soft-max	138	8.8
<b>VGG-16</b>																								
Image	Conv3-64	Conv3-64	Max pool	Conv3-128	Conv3-128	Max pool	Conv3-256	Conv3-256	Conv3-256	Conv3-256	Max pool	Conv3-512	Conv3-512	Max pool	Conv3-512	Conv3-512	Conv3-512	Max pool	FC-4096	FC-4096	FC-1000	Soft-max	144	9.0
<b>VGG-19</b>																								

# CNN Model

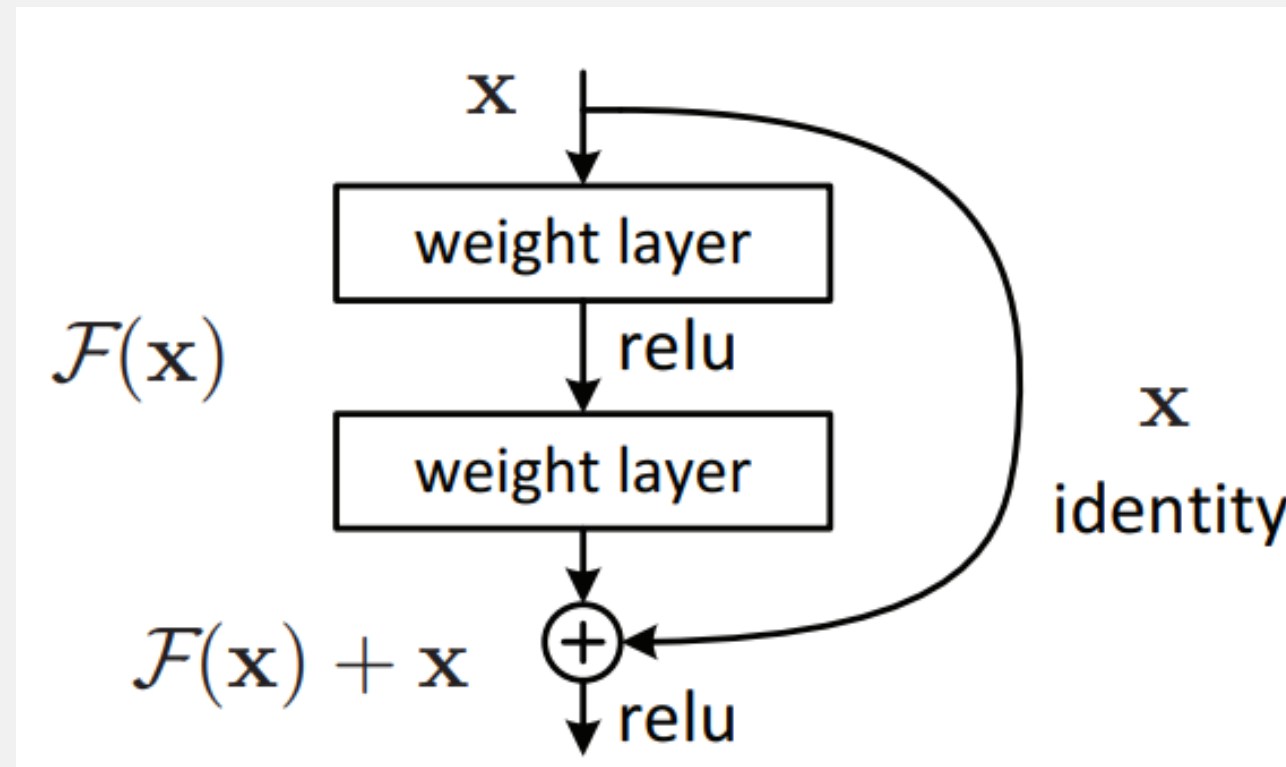


Inception

# CNN Model

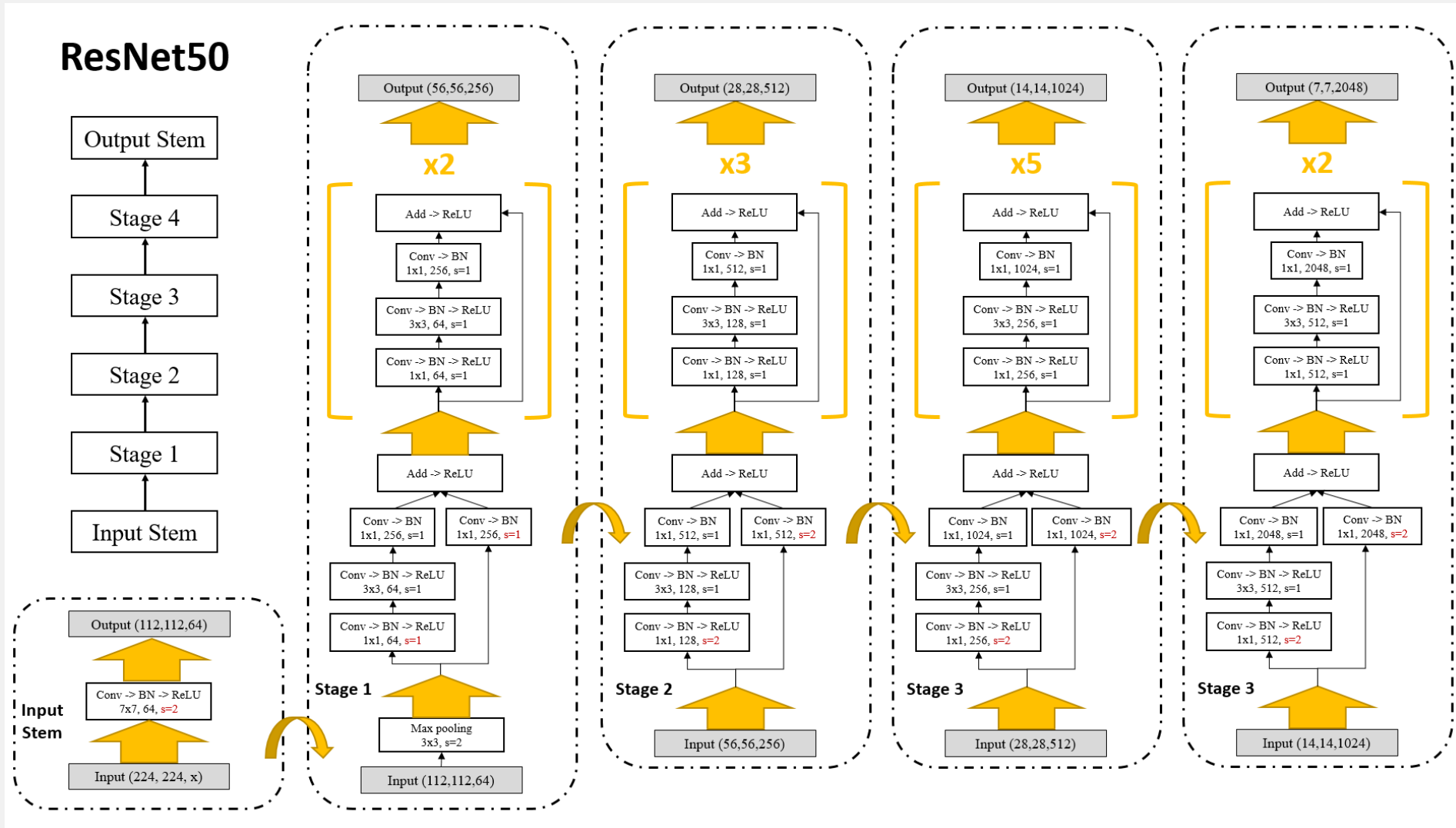
## Deep Residual Learning for image Recognition

➤ Cited: > 180,000



[ResNet](#)

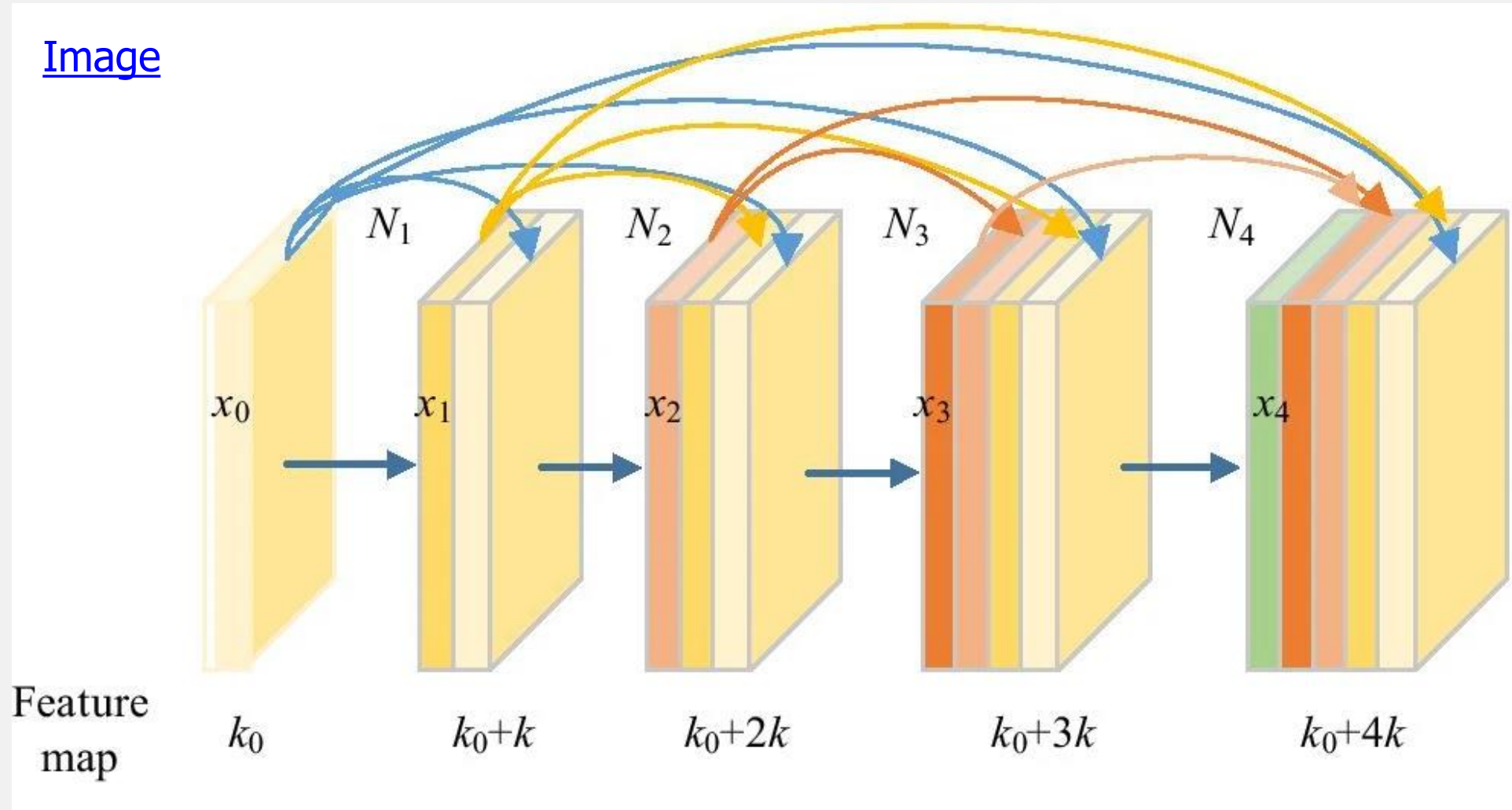
# CNN Model



[ResNet-50](#)

# CNN Model

Image



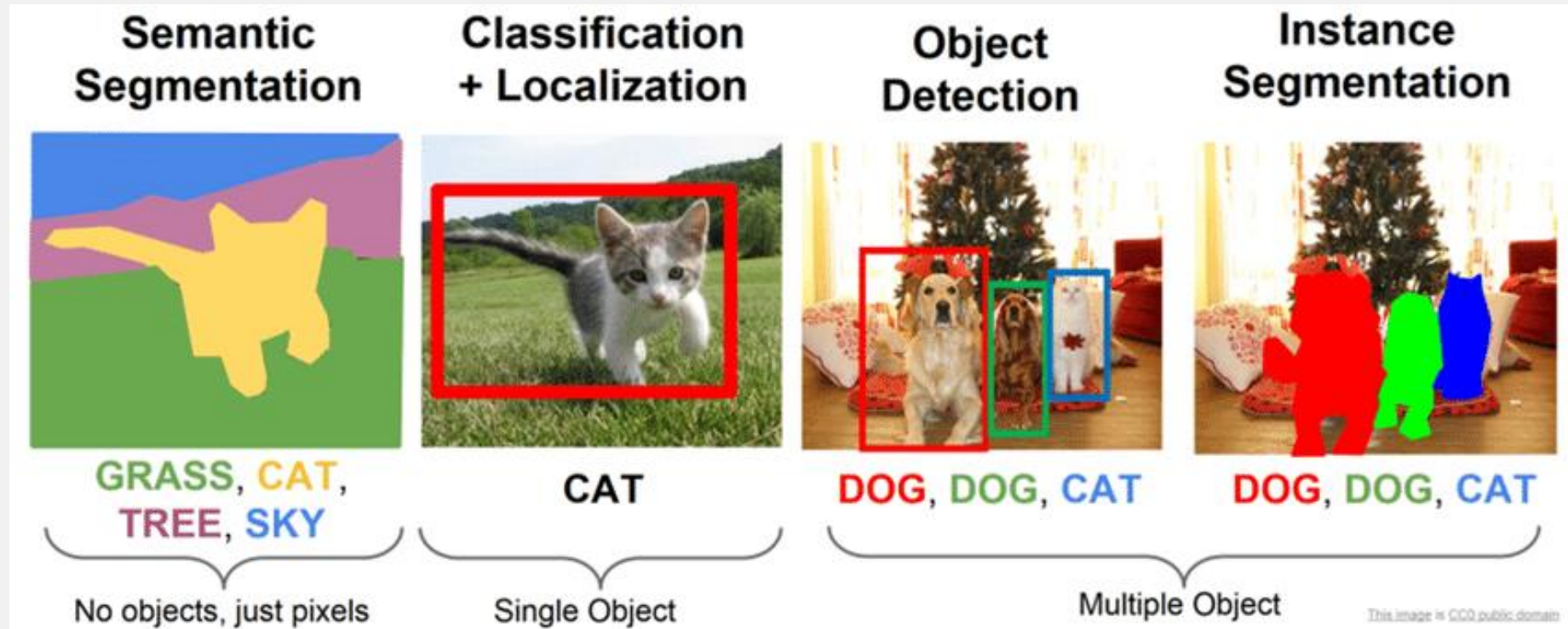
DenseNet



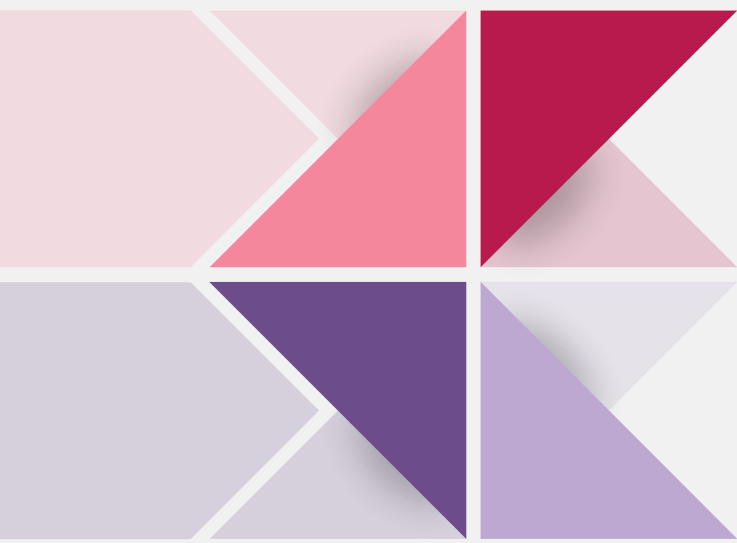
# CNN Model

## In Image

- Detection
- Segmentation
- Classification



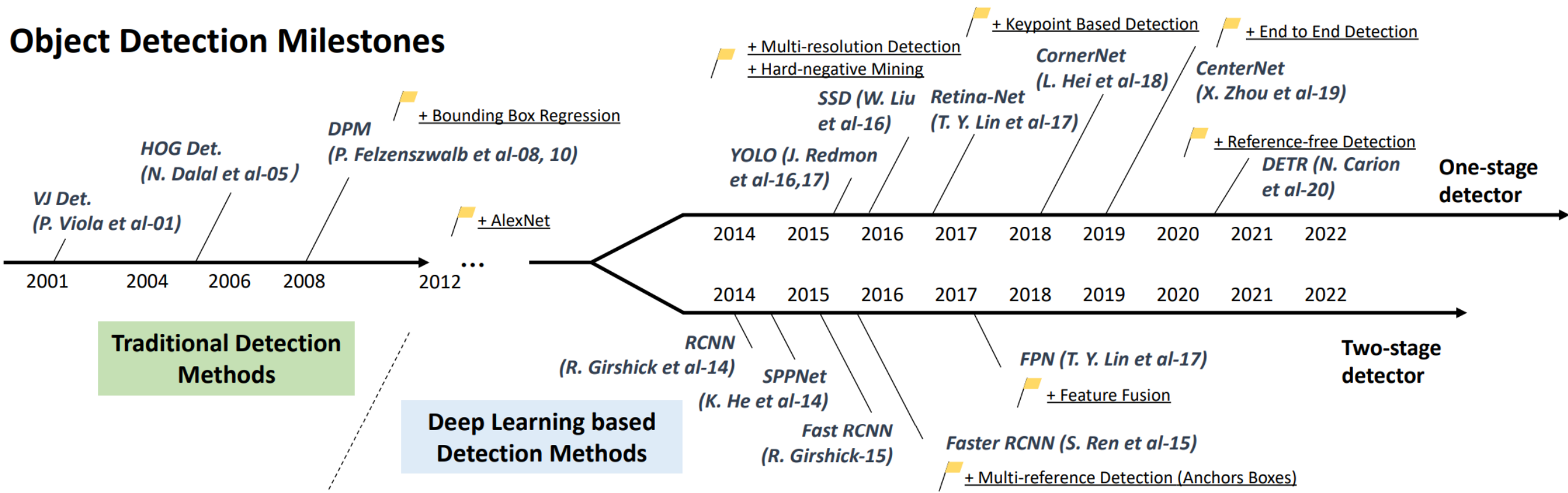
[Reference](#)



**01**

**Detection**

# Object Detection Milestones



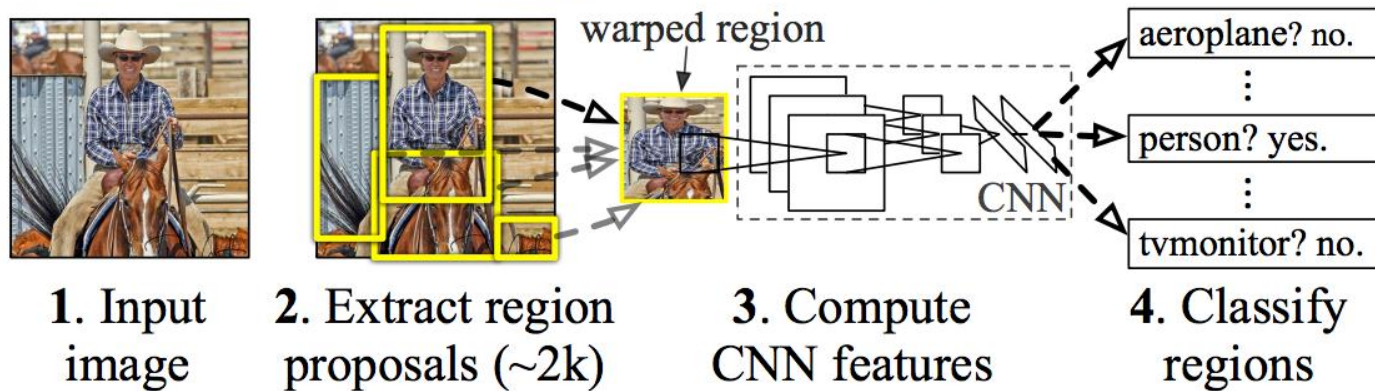
# CNN Model

## Object Detection

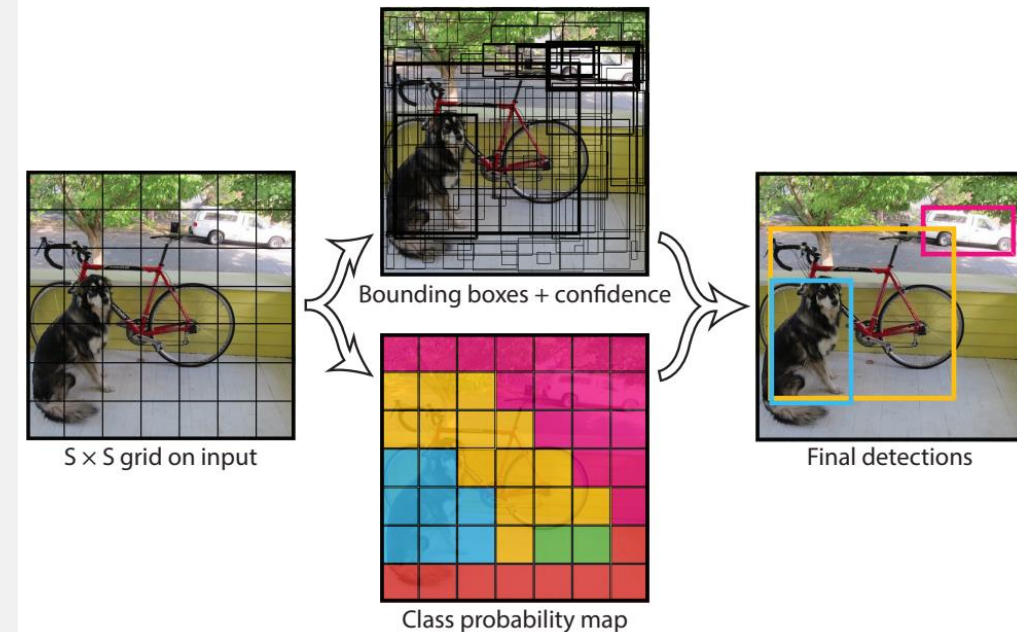
- R-CNN architecture (Two-stage)
- YOLO architecture (One-stage)

### R-CNN

#### **R-CNN: Regions with CNN features**

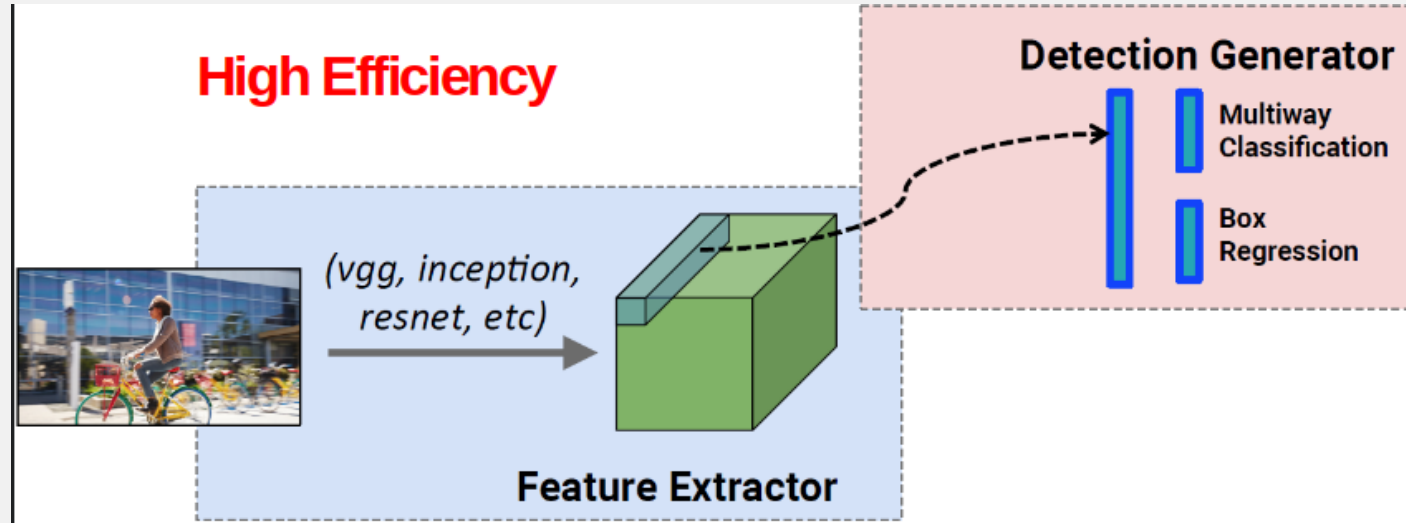


### YOLO

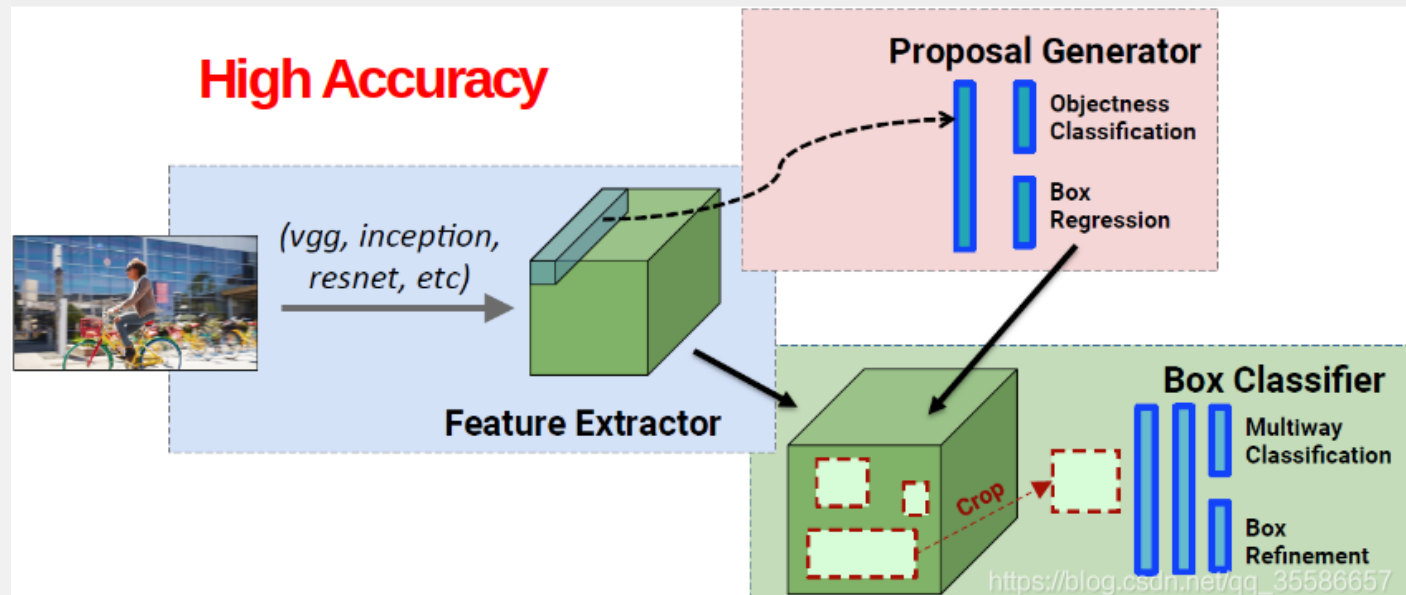


# CNN Model

One-stage



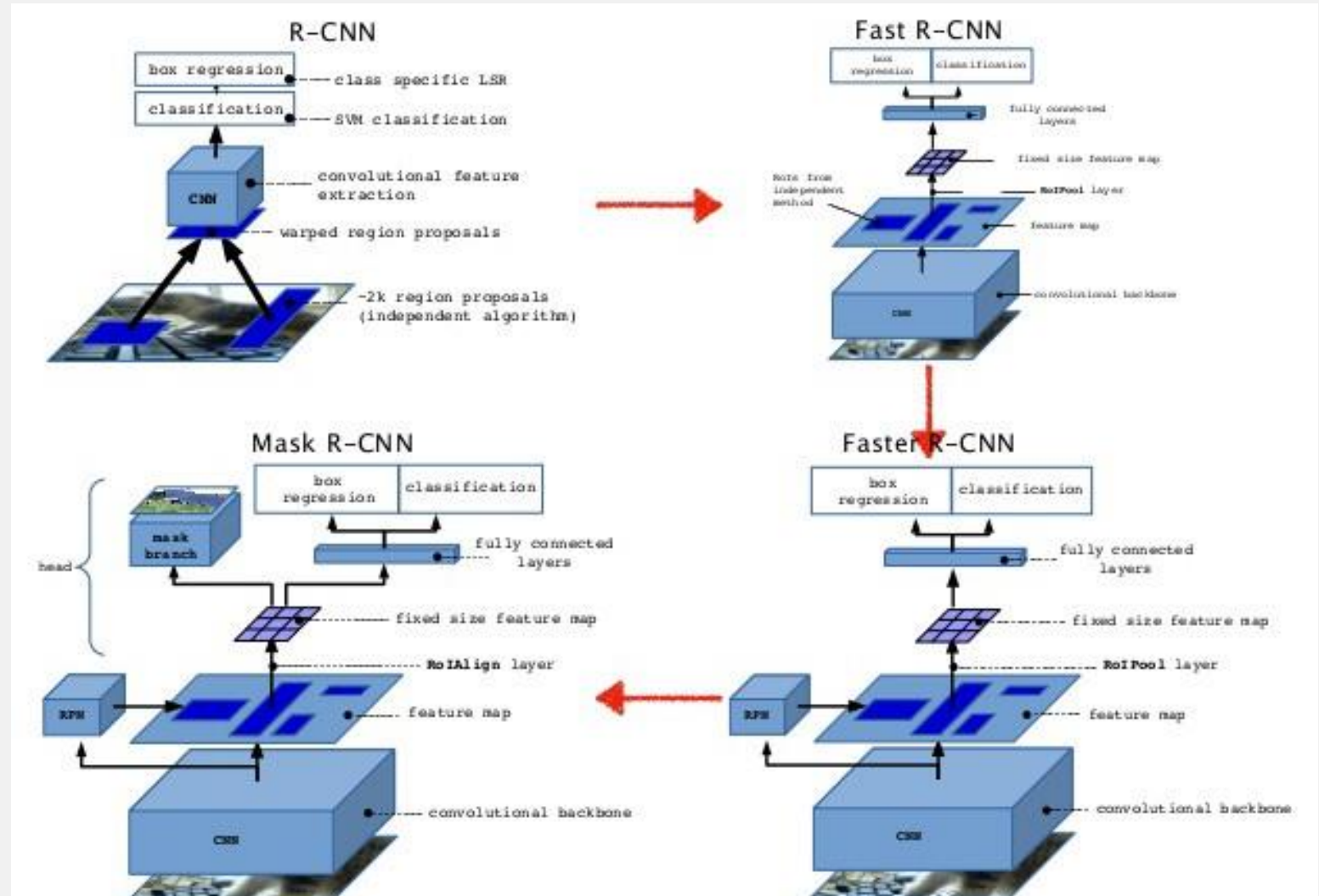
Two-stage



# CNN Model

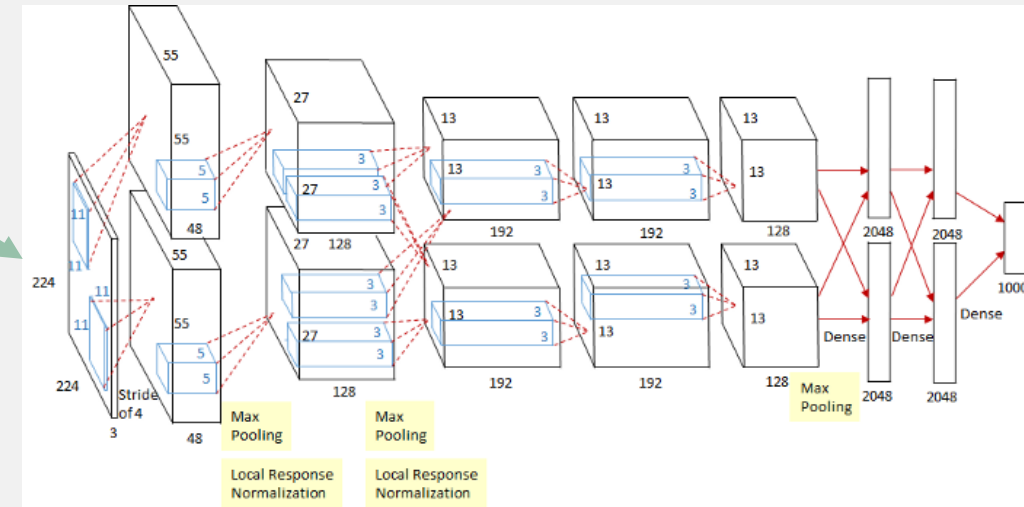
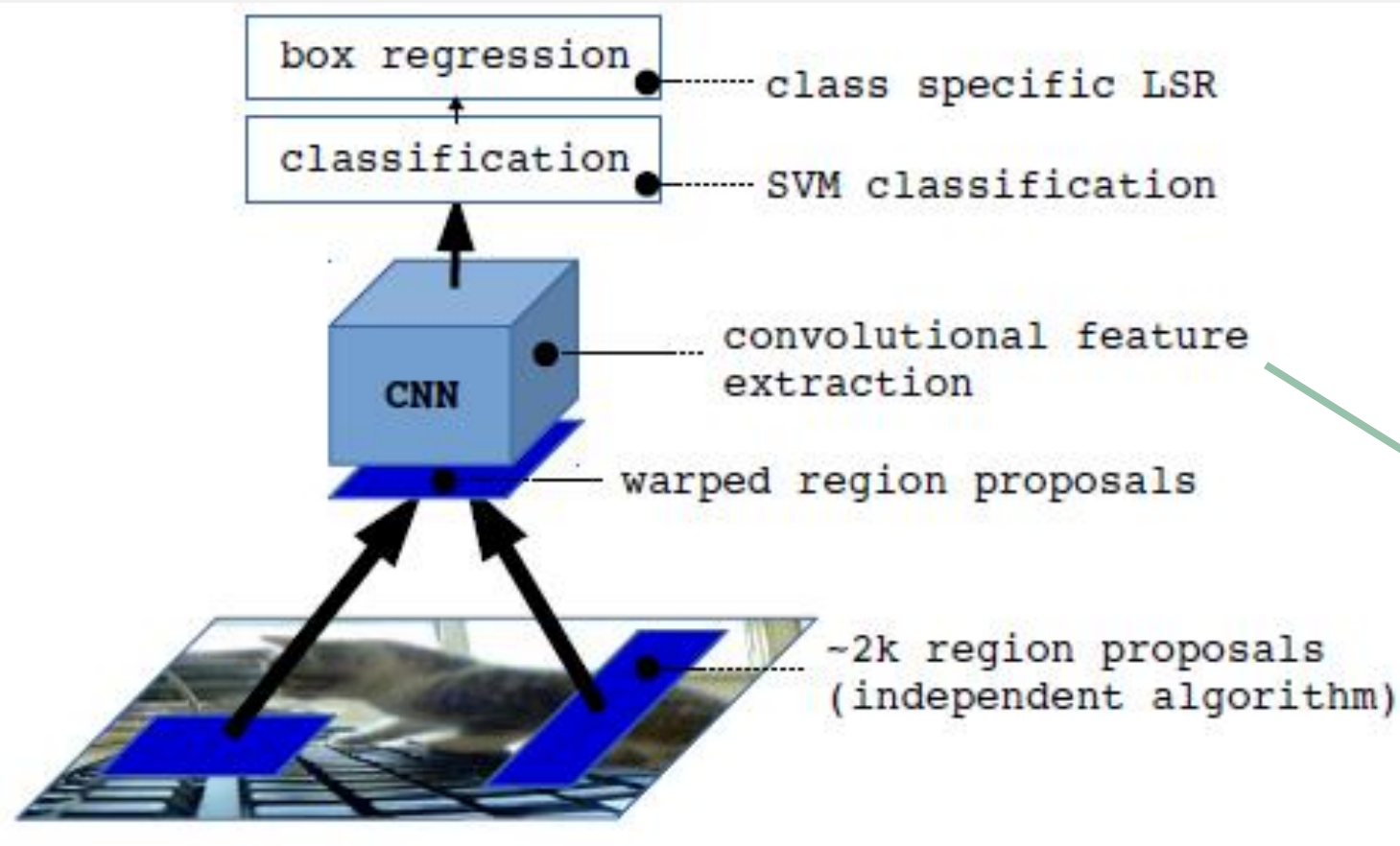
## R-CNN family

- R-CNN
- Fast R-CNN
- Faster R-CNN
- Mask R-CNN



# CNN Model

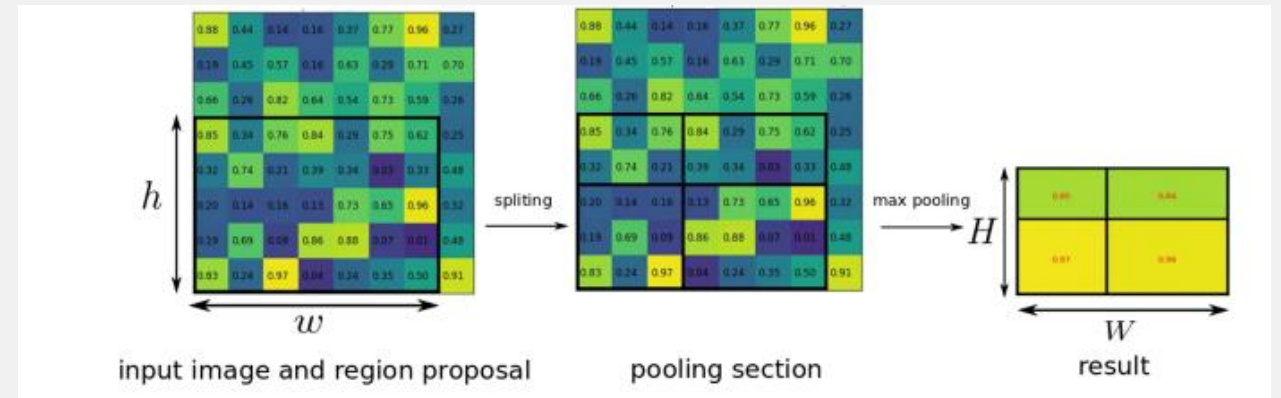
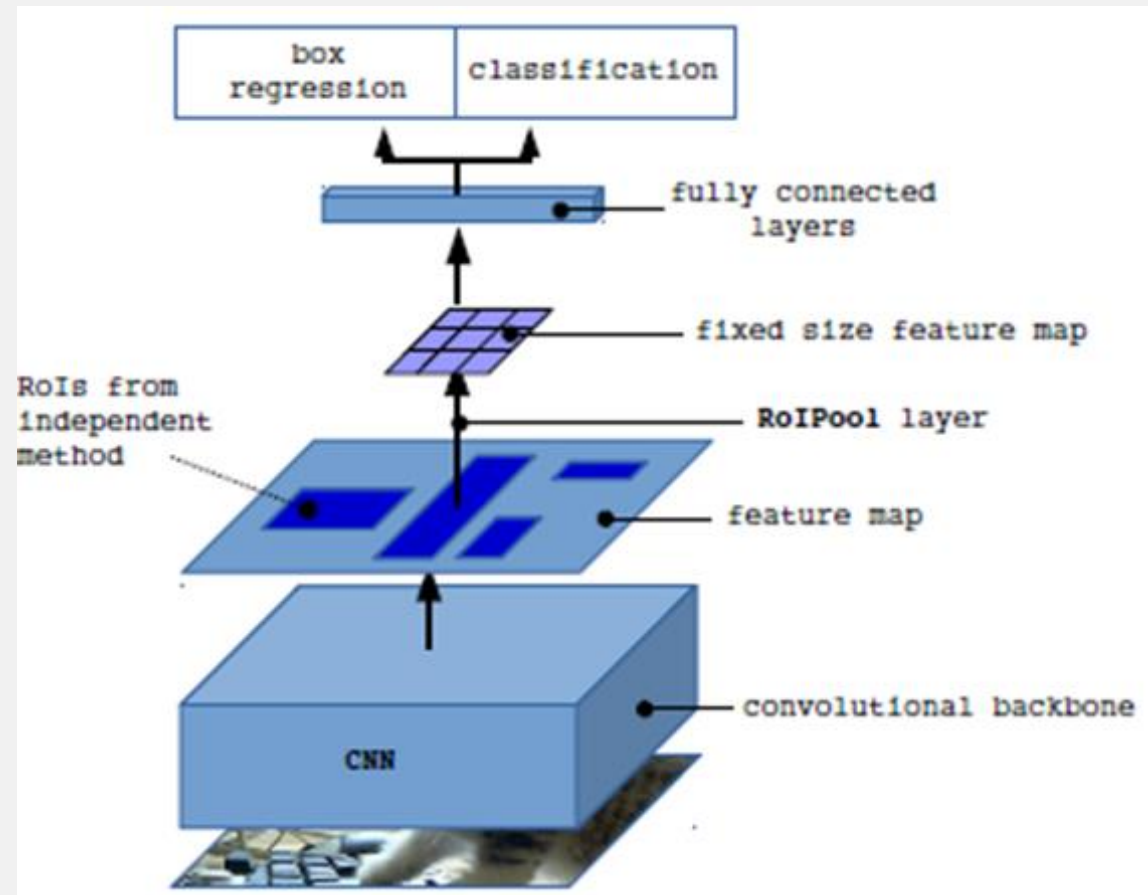
## R-CNN



R-CNN

# CNN Model

## Fast-CNN

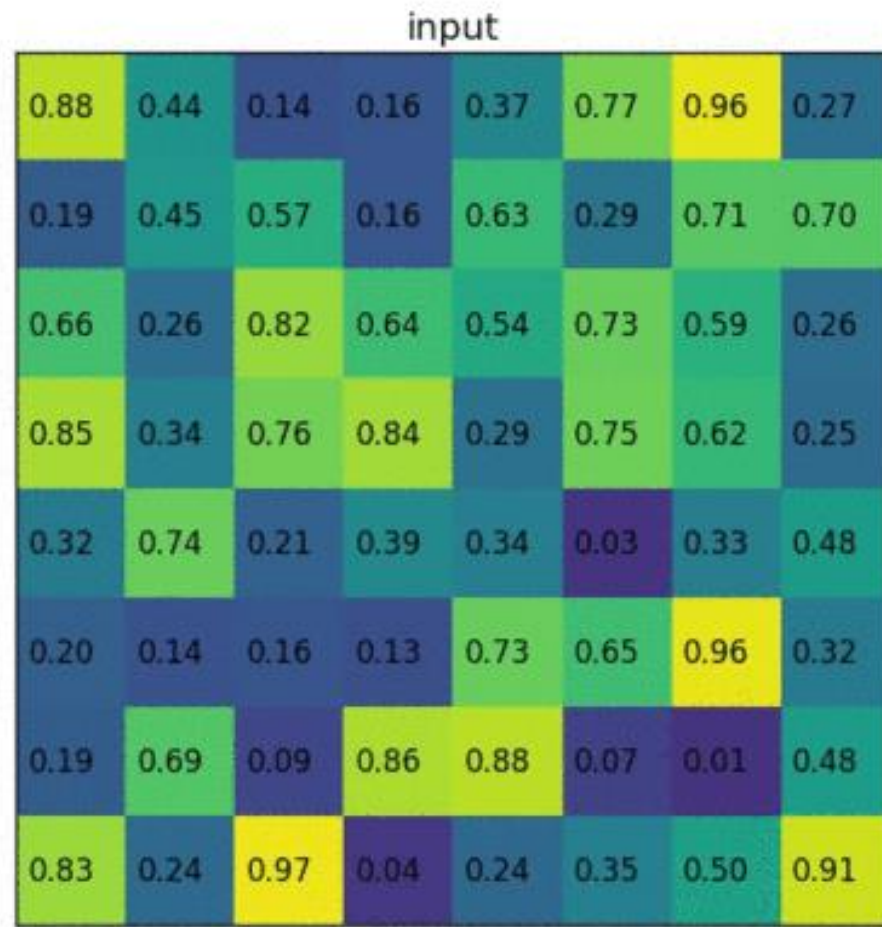


[ROI Pooling](#)

[Fast R-CNN](#)

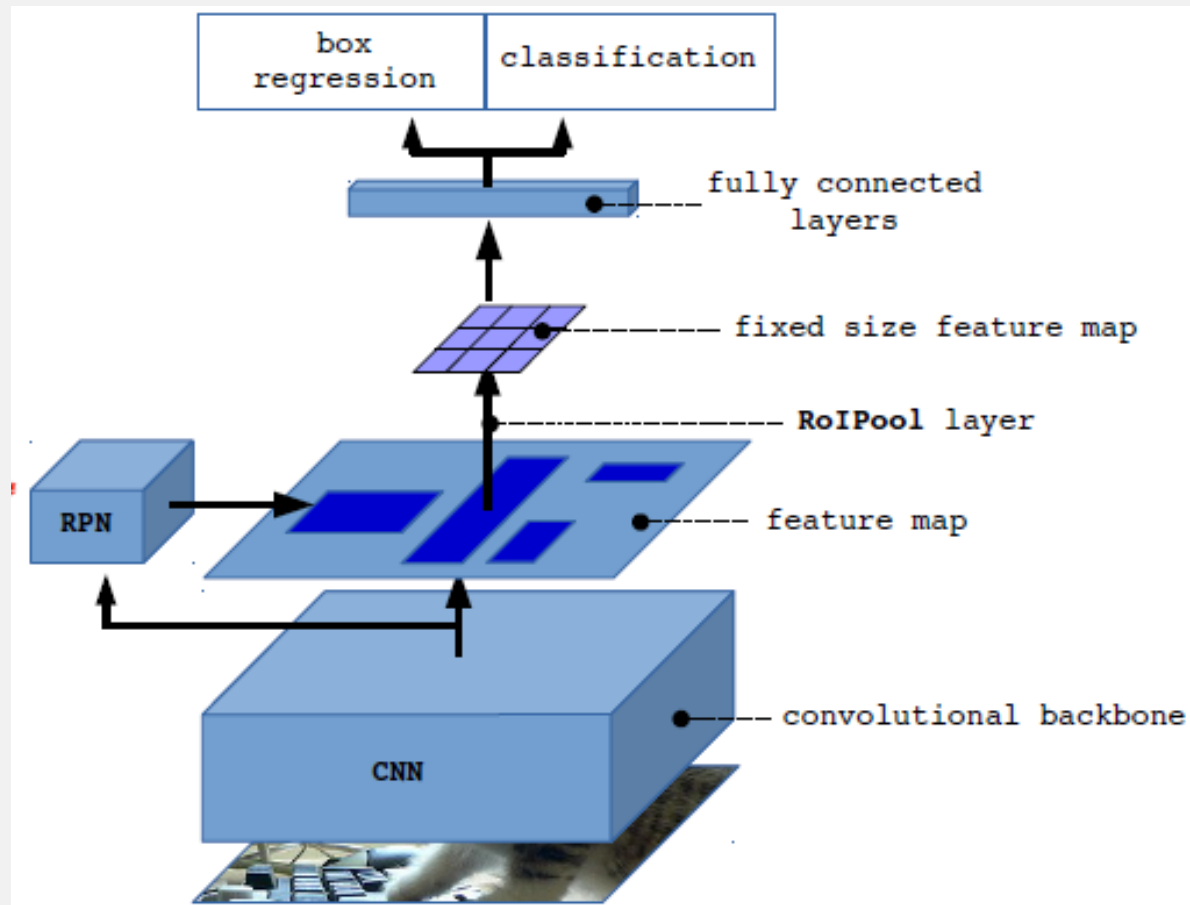


# CNN Model

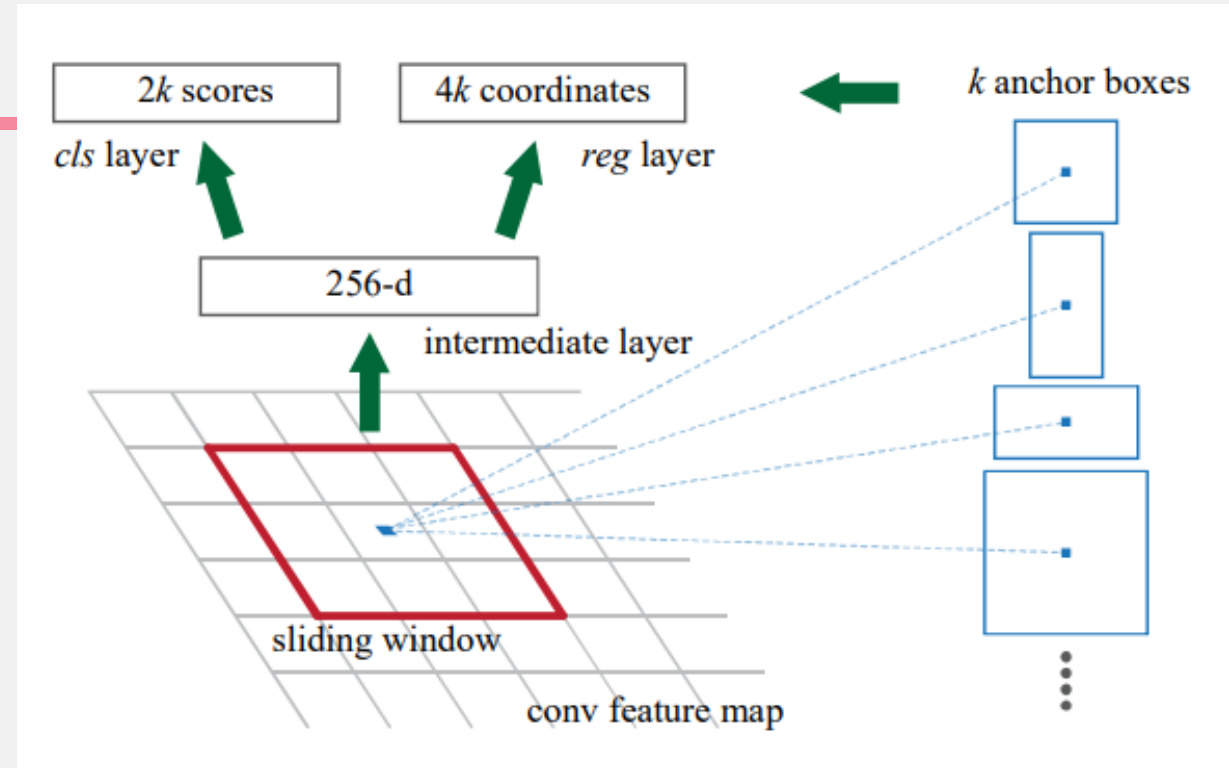


# CNN Model

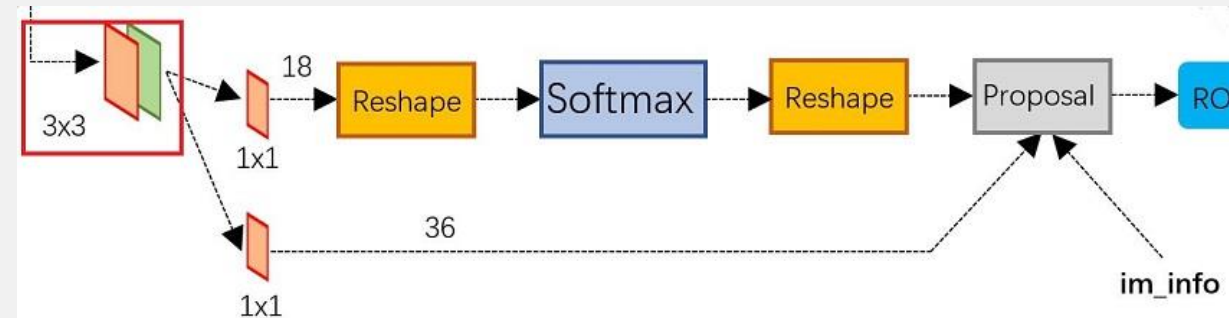
## Faster-CNN



Faster R-CNN



RPN and Anchor



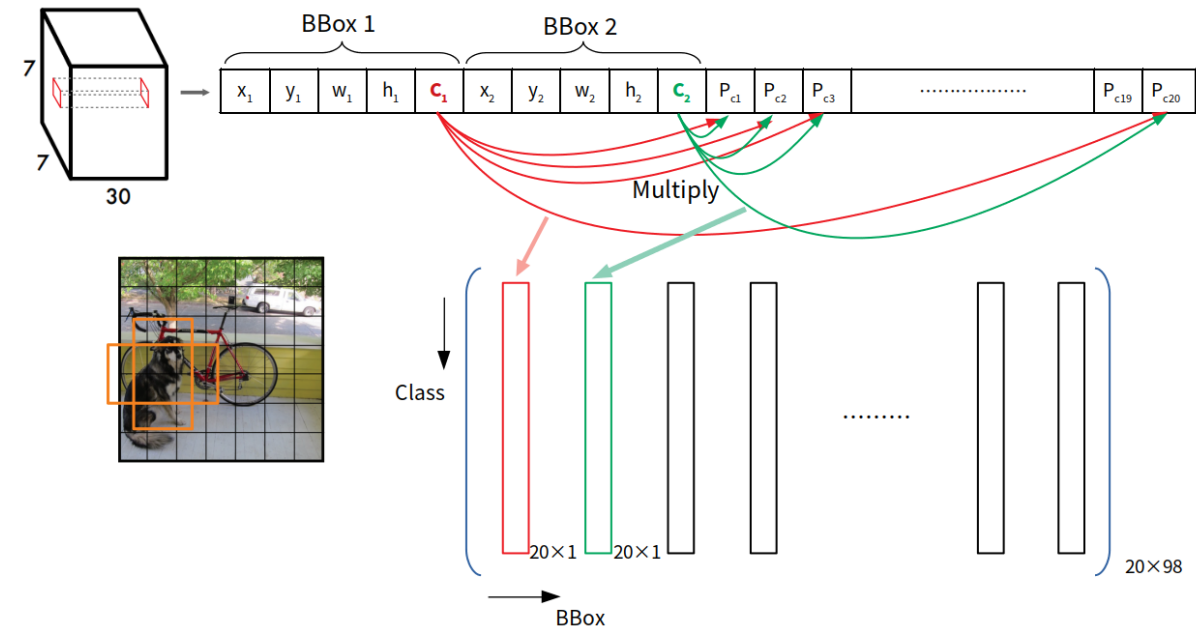
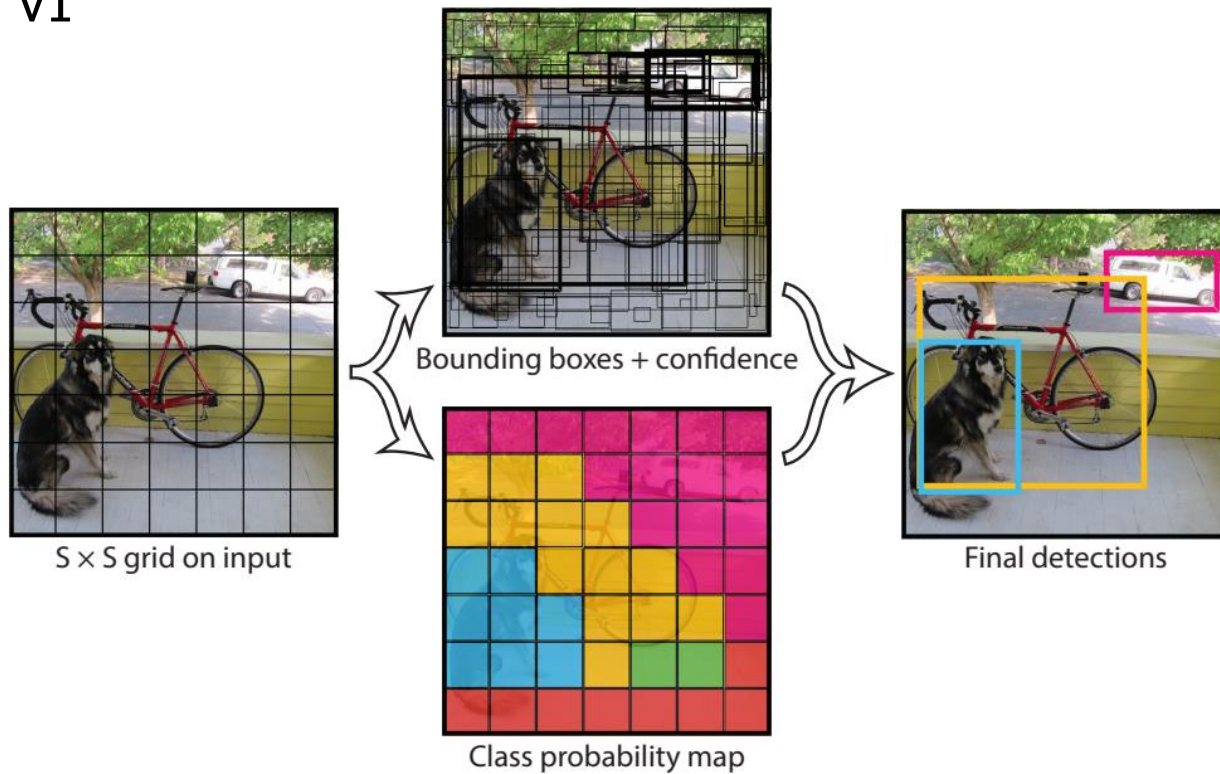
RPN

# CNN Model

## YOLO (You Only Look Once)

➤ v8 is the latest

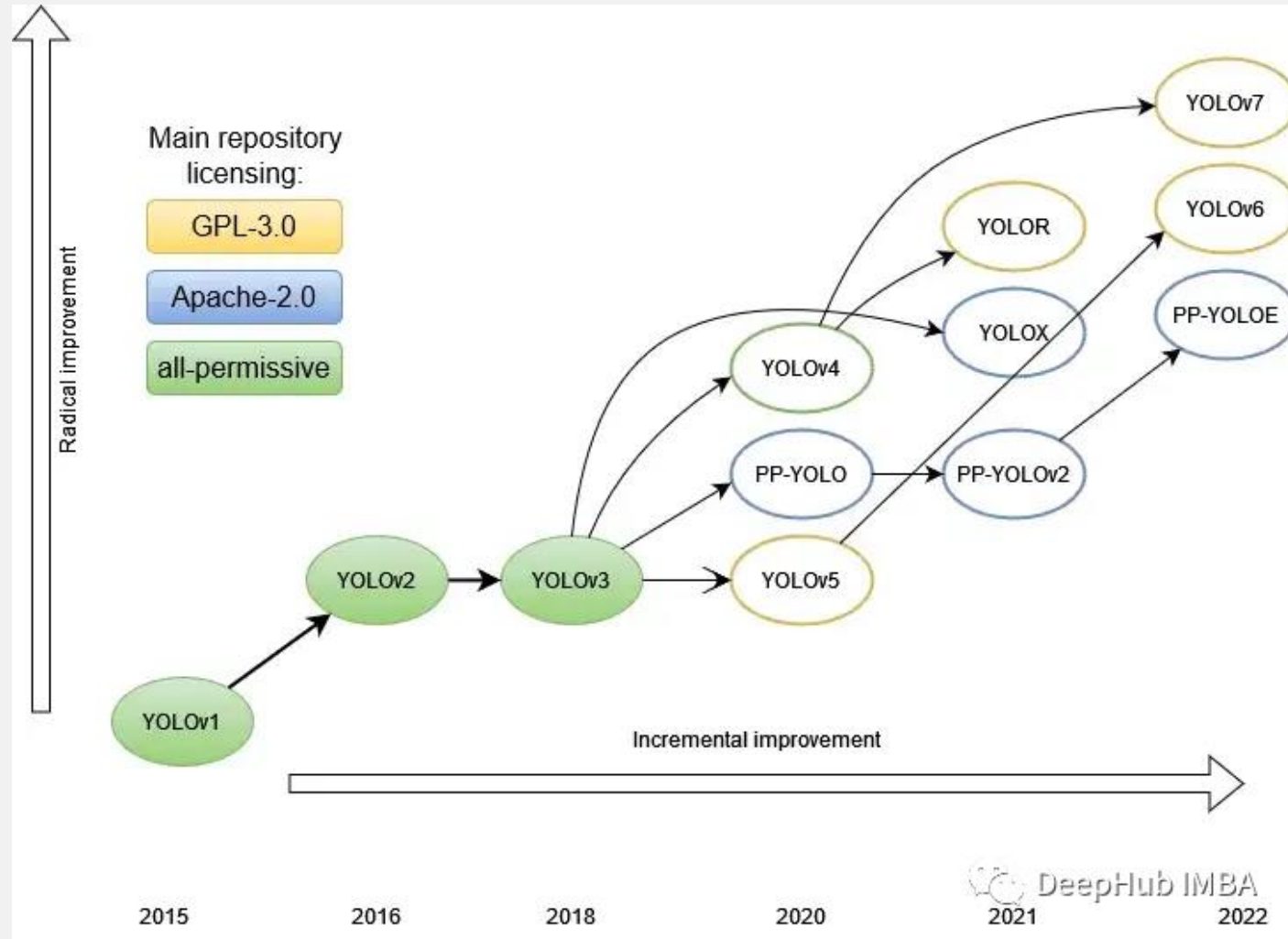
v1




[Reference](#)

# CNN Model

## YOLO Genealogy



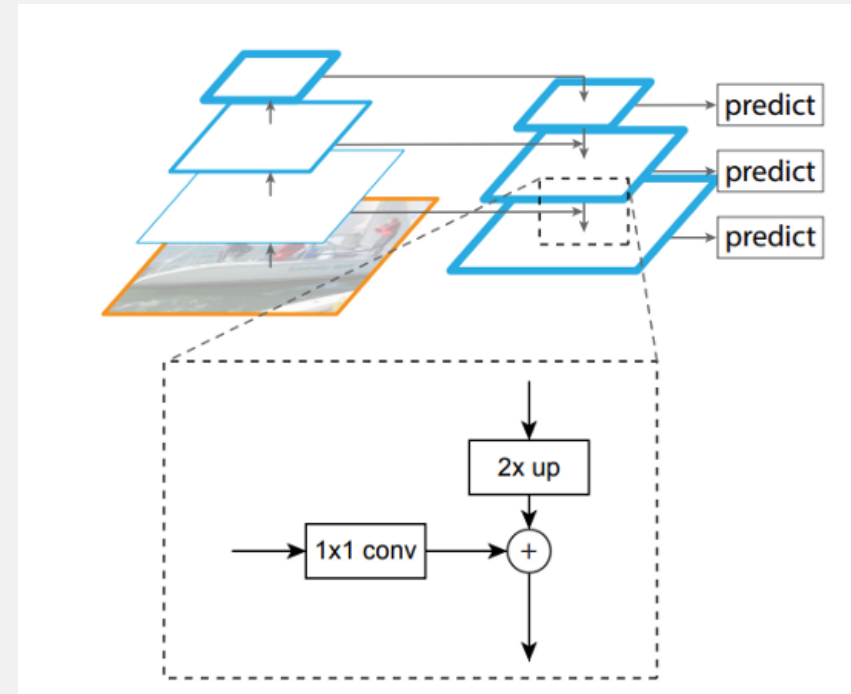
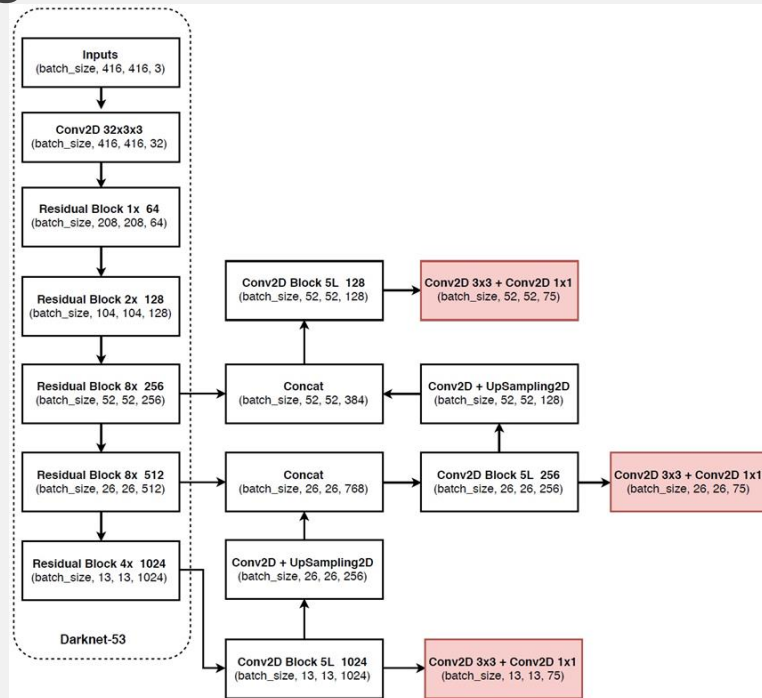
YOLO version	backbone	neck	head(s)	augmentations
YOLOv1	GoogLeNet, VGG-16	2x fully connected layers	combined classes + bboxes	random scaling & translations up to 20%; random adjust exposure & saturation up to x1.5 in HSV
YOLOv2	Darknet-19	fully convolutional layers	combined hierarchical classes + bboxes, anchor-based	random crops, rotations, and hue, saturation, and exposure shifts
YOLOv3	Darknet-53	FPN	combined multilabel + bboxes, anchor-based	no specific info, seems like the same as in YOLOv2
YOLOv5	CSPDarknet53	SPPF, CSP-PAN	combined multilabel + bboxes, anchor-based	Mosaic, copy-paste, random affine, MixUp, random adjust HSV, random horizontal flip
PP-YOLO	ResNet50-vd + deformable convolutions	FPN, SPP	combined multilabel + bboxes, anchor-based	MixUp
YOLOv4	CSPDarknet53	PANet, SPP	combined multilabel + bboxes, anchor-based	CutMix, Mosaic, MixUp, CutOut, Self-Adversarial Training, bilateral blurring
PP-YOLOv2	ResNet50-vd + deformable convolutions	PANet	combined multilabel + bboxes, anchor-based	MixUp; random color distortion, expand, crop, flip
YOLOX	Darknet-53	FPN	decoupled multilabel + bboxes, anchor-free	Mosaic, MixUp, random horizontal flip, colorjitter
YOLOR	sequence of convolutional layers with downscaling	FPN, CSP, SPP	multi-head (object detection, multi-label classification, feature embedding)	CutMix, Mosaic, MixUp, CutOut, Self-Adversarial Training, bilateral blurring
PP-YOLOE	CSPRepResNet	PANet	Efficient Task-aligned Head (decoupled), anchor-free	random crop, horizontal flip, color distortion, multi-scale
YOLOv6	EfficientRep	Rep-PAN	Efficient decoupled head, anchor-free	Mosaic, 
YOLOv7	Extended-ELAN	-	multiple (lead heads & aux heads), anchor-based	random perspective, HSV jitter, flips, Mosaic

## [Reference](#)

# CNN Model

## YOLOv3

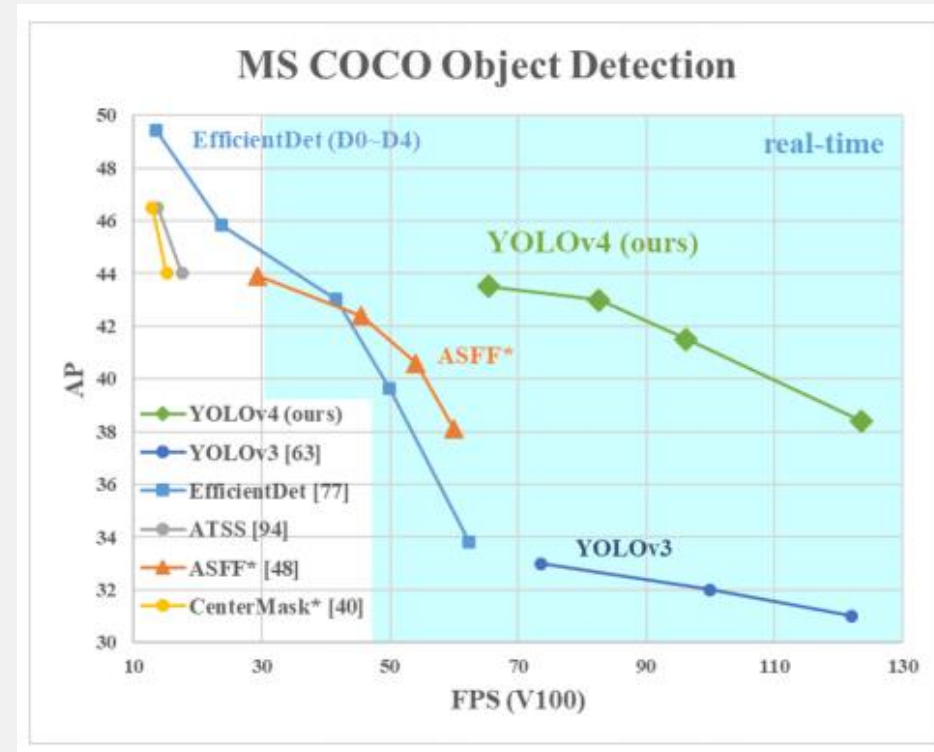
- Using Darknet-53 for feature extraction (ResNet block)
- Using FPN (Feature Pyramid Network)
- Using Anchor



# CNN Model

## YOLOv4

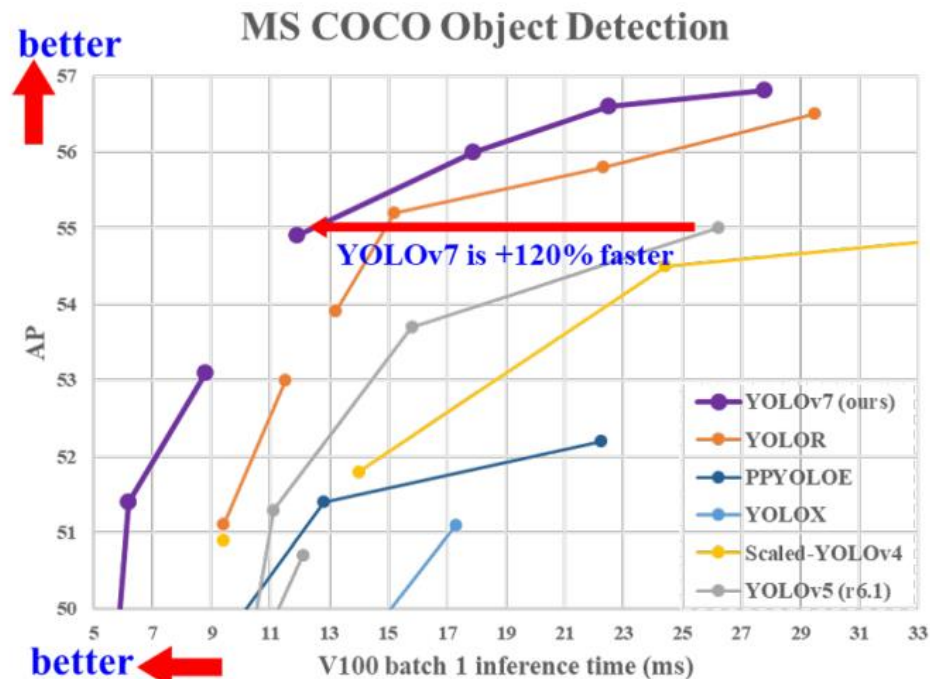
- Using CSPDarknet-53 for feature extraction
- Using PANet + SPP
- Using CmBN (Cross mini-Batch Normalization)
- Training Method
  - Bag-of-Freebies and Bag-of-Specials



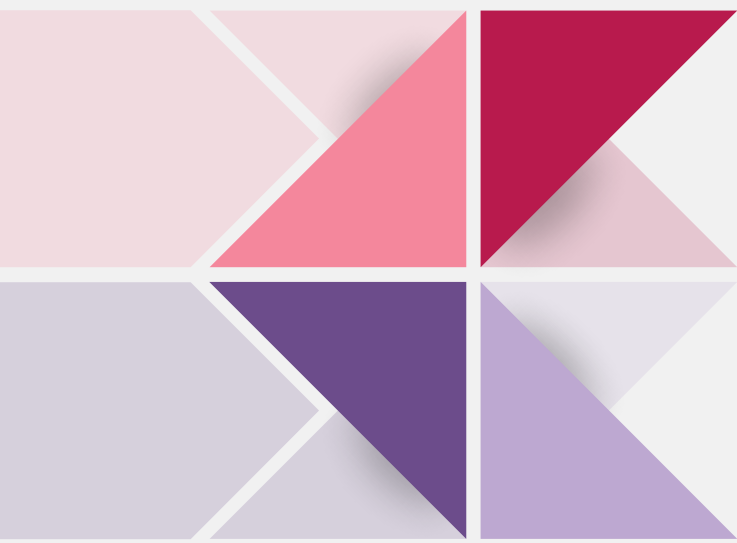
# CNN Model

## YOLOv7

- Extended efficient layer aggregation networks
- Model scaling
- Auxiliary and lead head



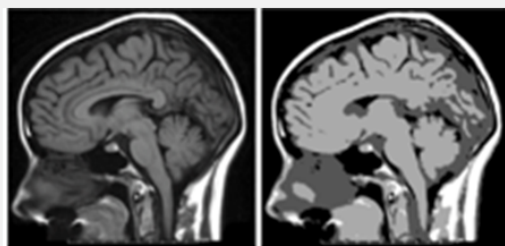




**02**

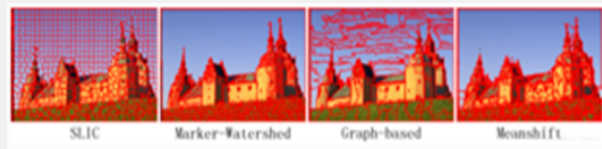
**Segmentation**

# CNN Model



## 第一代 閾值演算法(~1960)

Otsu, Nobuyuki. "A threshold selection method from gray-level histograms." *IEEE*



## 第二代 邊緣區域輪廓(~1970)

分水嶺, K-means, 邊緣檢測, opencv方法



## 第三代 語意切割 & 實例分割

FCN, Mask RCNN, U-net,

深度學習

特徵工程

FCN(2014)

CNN崛起(2012)

像素值

微處理器、大規模硬體設備(~1970)

數位影像處理(~1960)

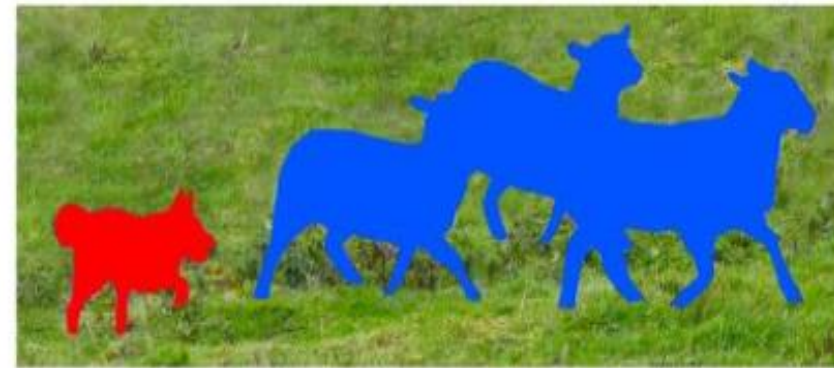
數位影像(1921)

# CNN Model

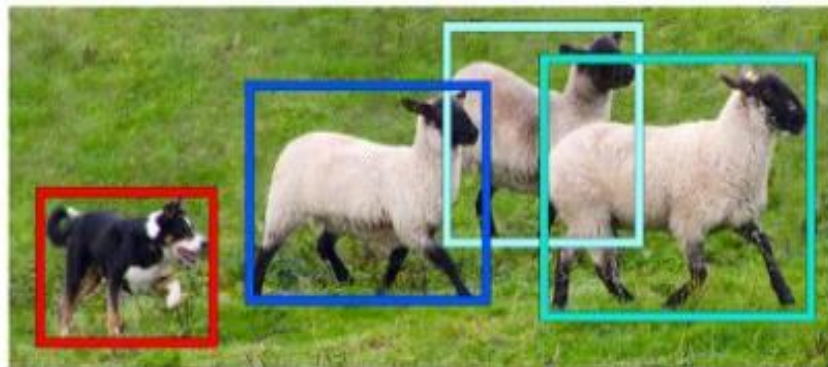
## Object Segmentation



Image Recognition



Semantic Segmentation



Object Detection

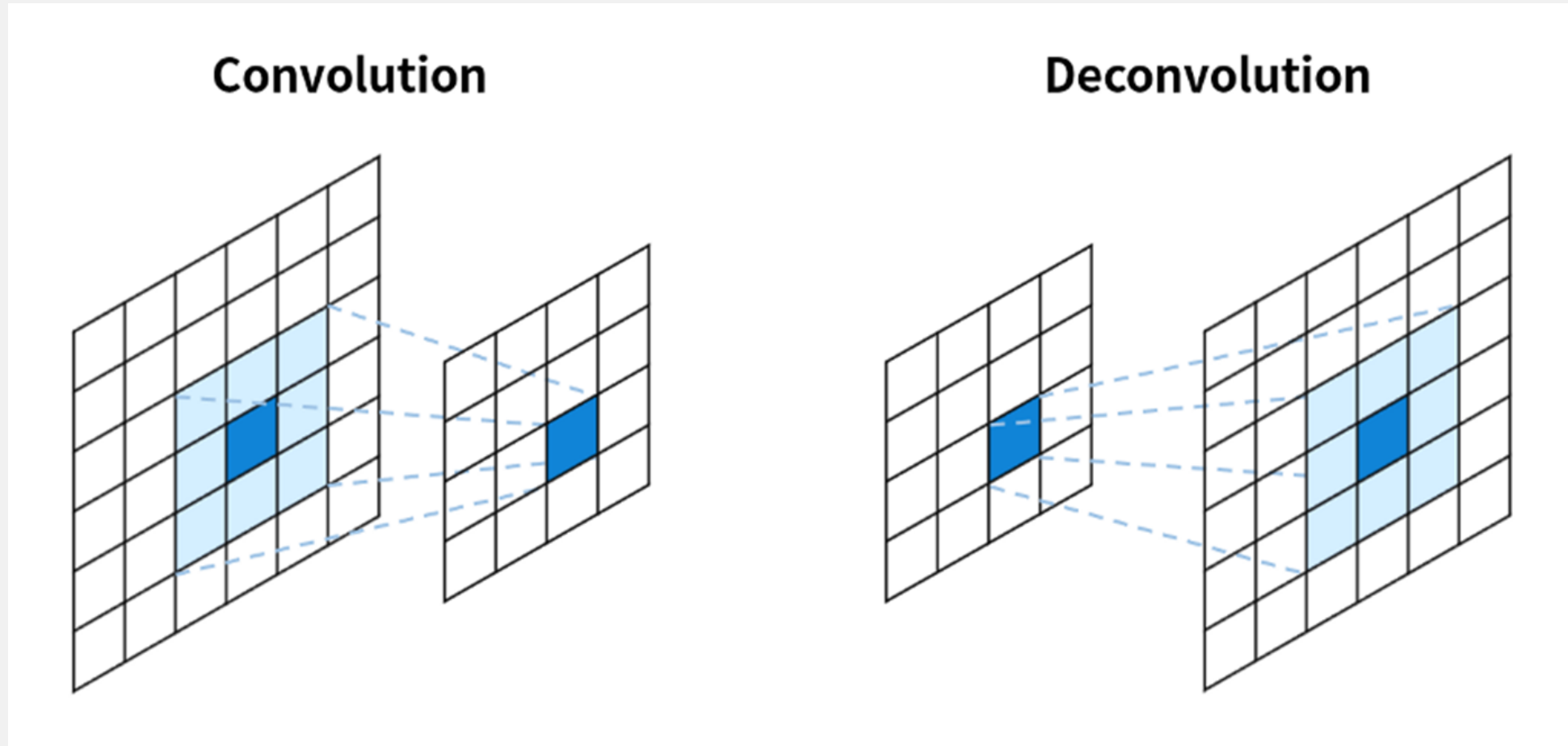


Instance Segmentation

# CNN Model

Semantic Segmentation

## Deconvolution

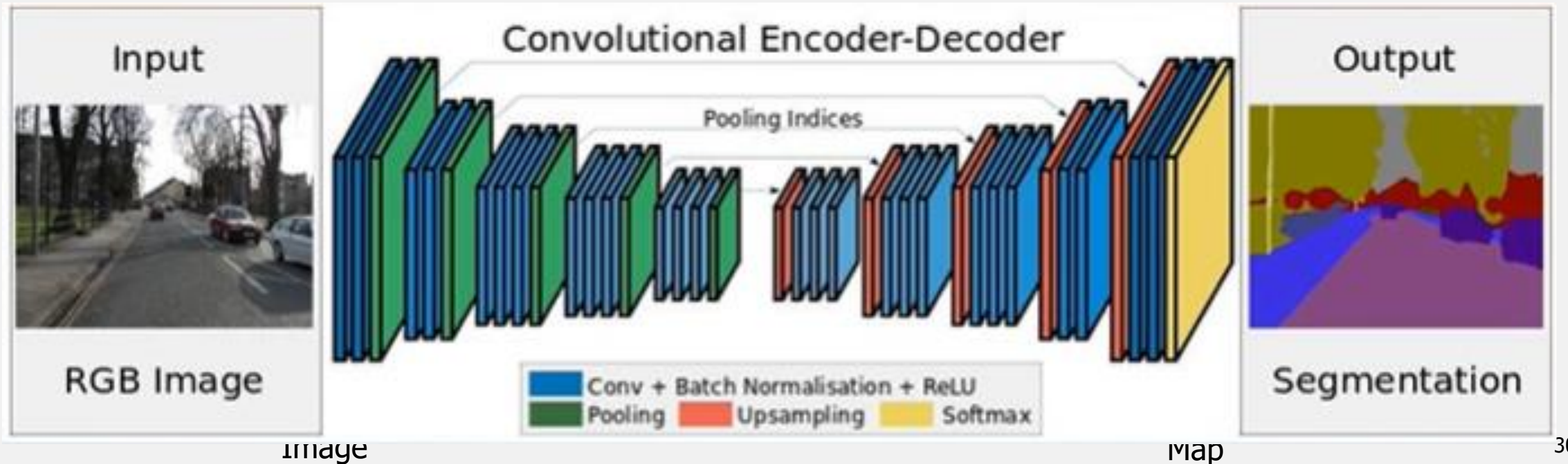


# CNN Model

## Semantic Segmentation

### U-Net

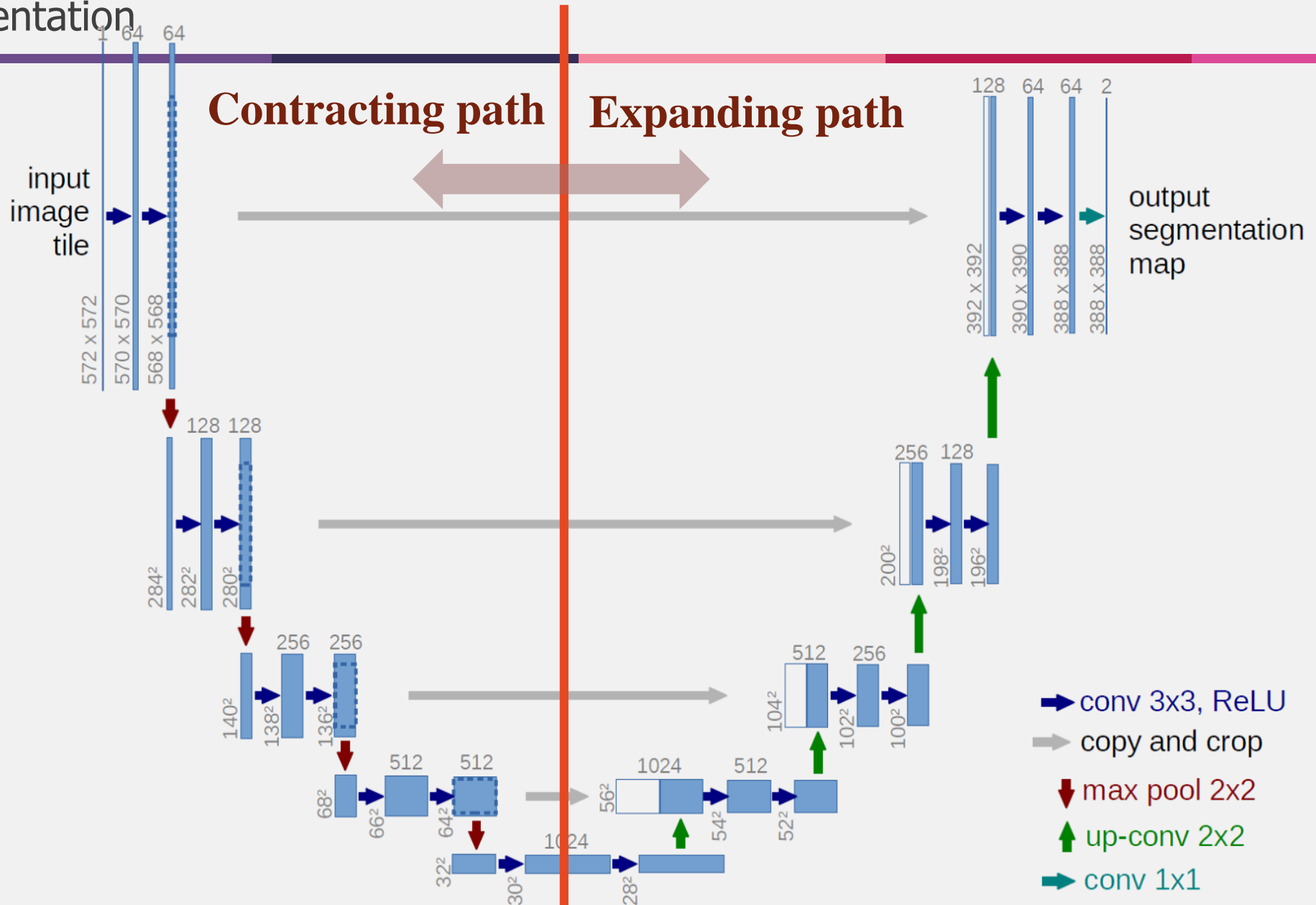
- Created in 2015, U-net was a unique CNN developed for Biomedical Image Segmentation
- U-net learns segmentation in an end-to-end setting, has now become a very popular end-to-end encoder-decoder network for semantic segmentation



# CNN Model

## Semantic Segmentation

### U-Net

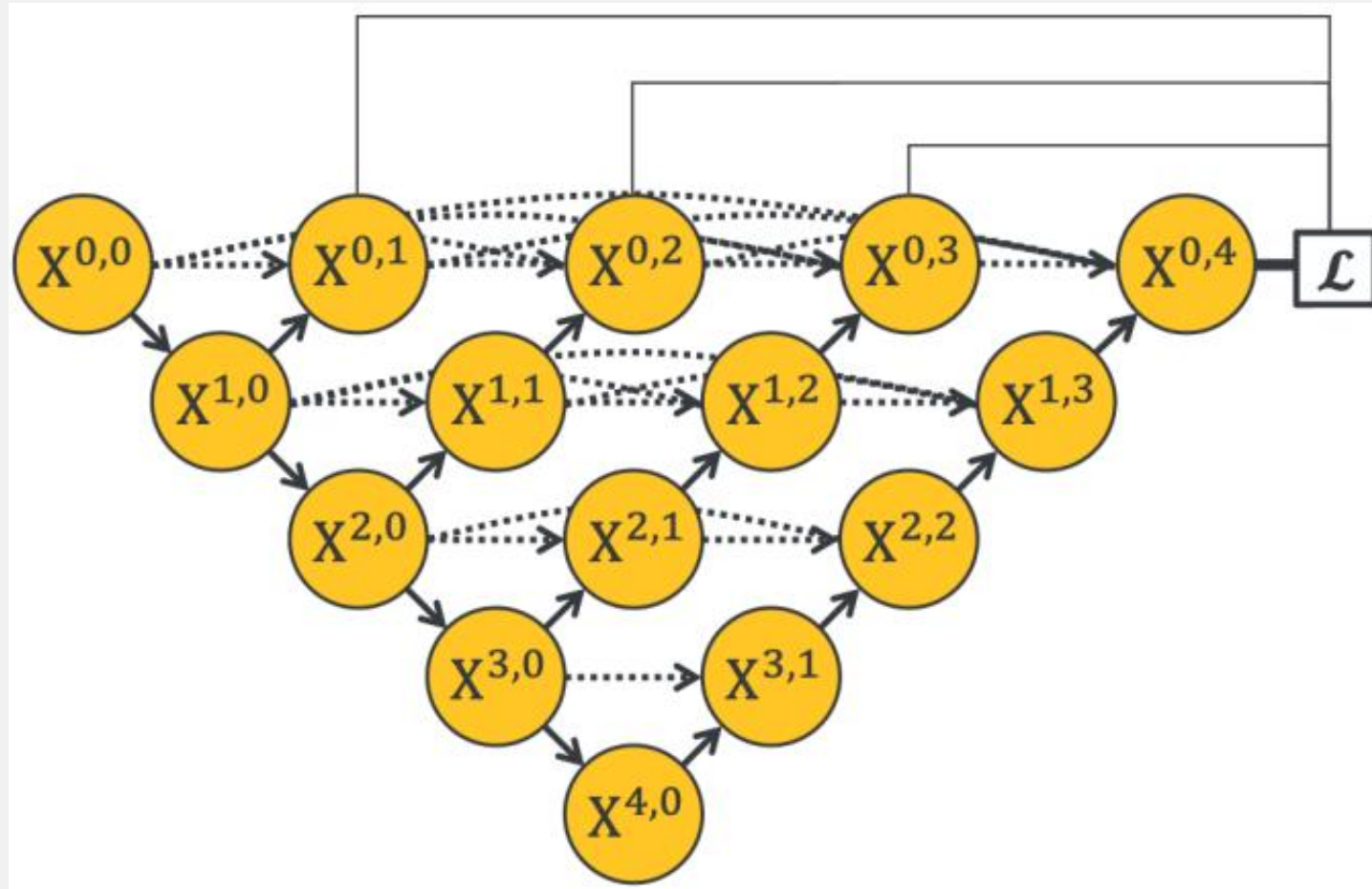


[Reference](#)

# CNN Model

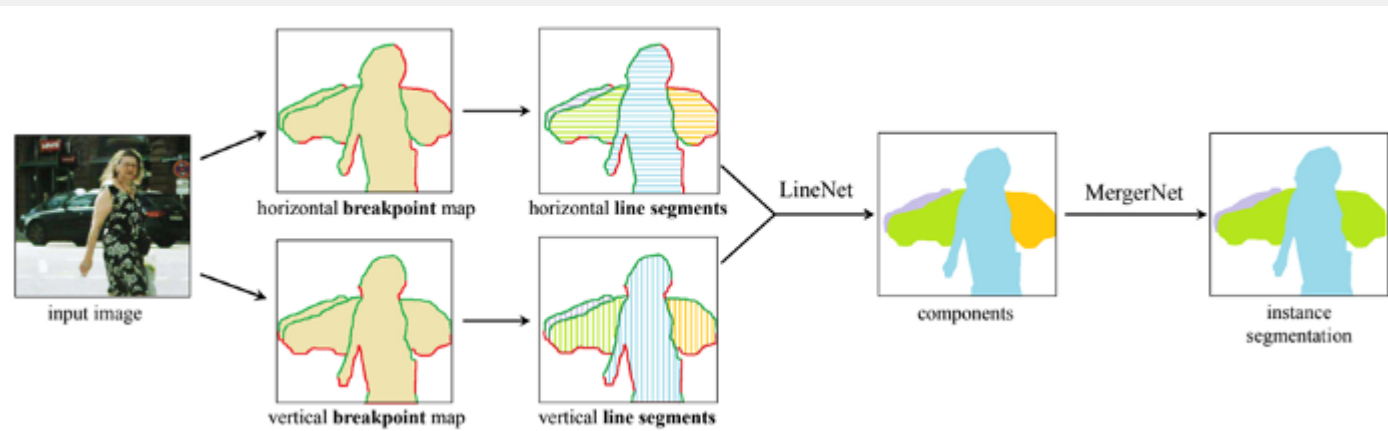
Semantic Segmentation

U-Net++

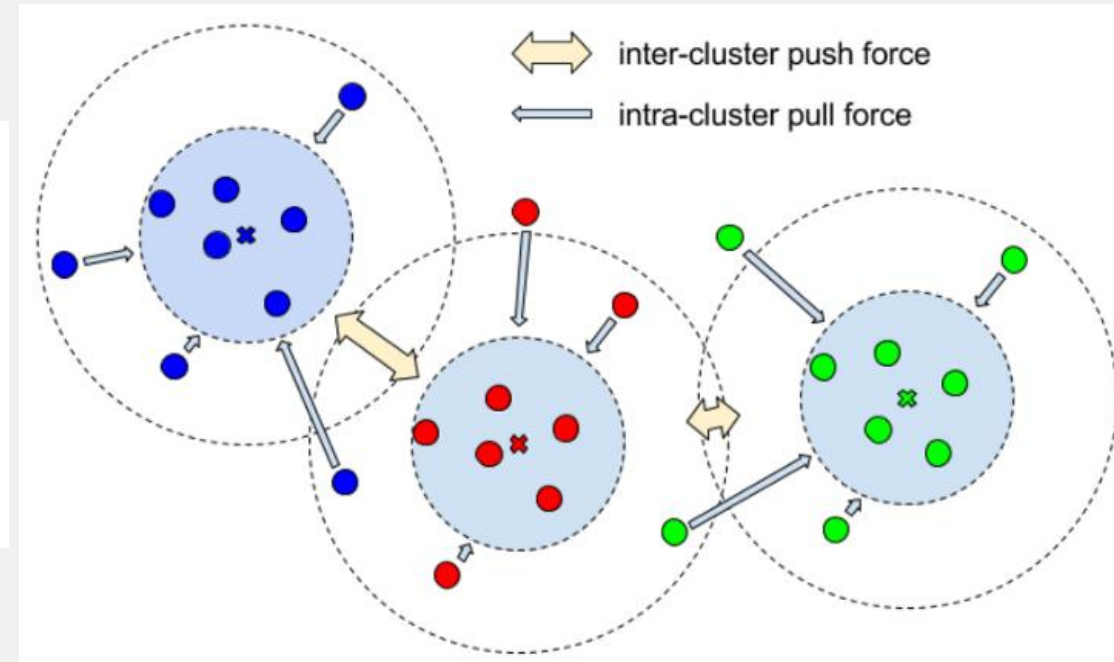


# CNN Model

## Instance Segmentation



[SGN](#)

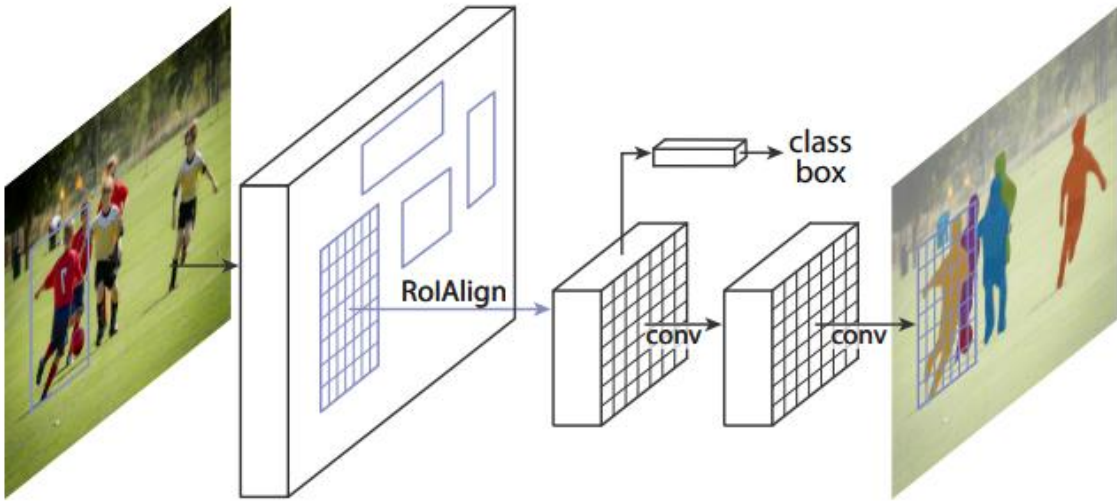


[Discriminative Loss](#)

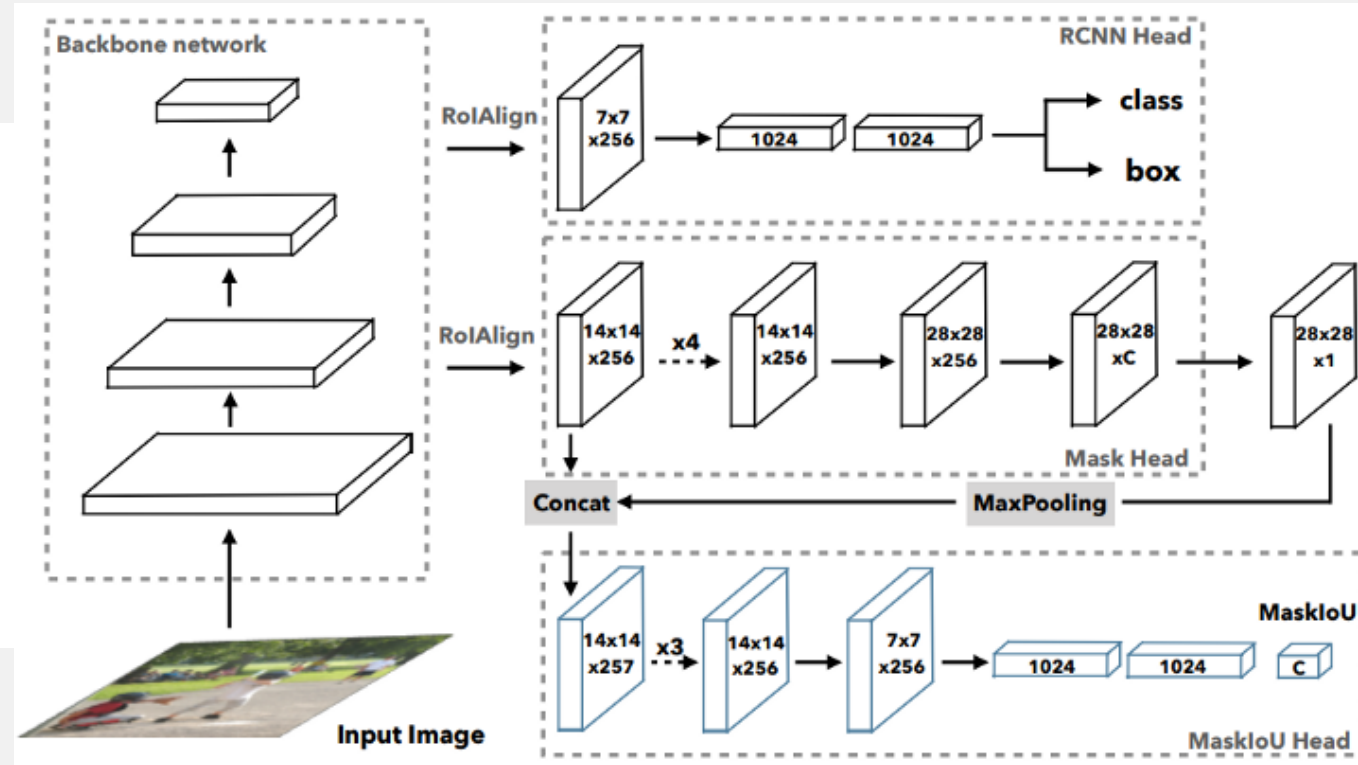


# CNN Model

## Instance Segmentation



[Mask R-CNN](#)



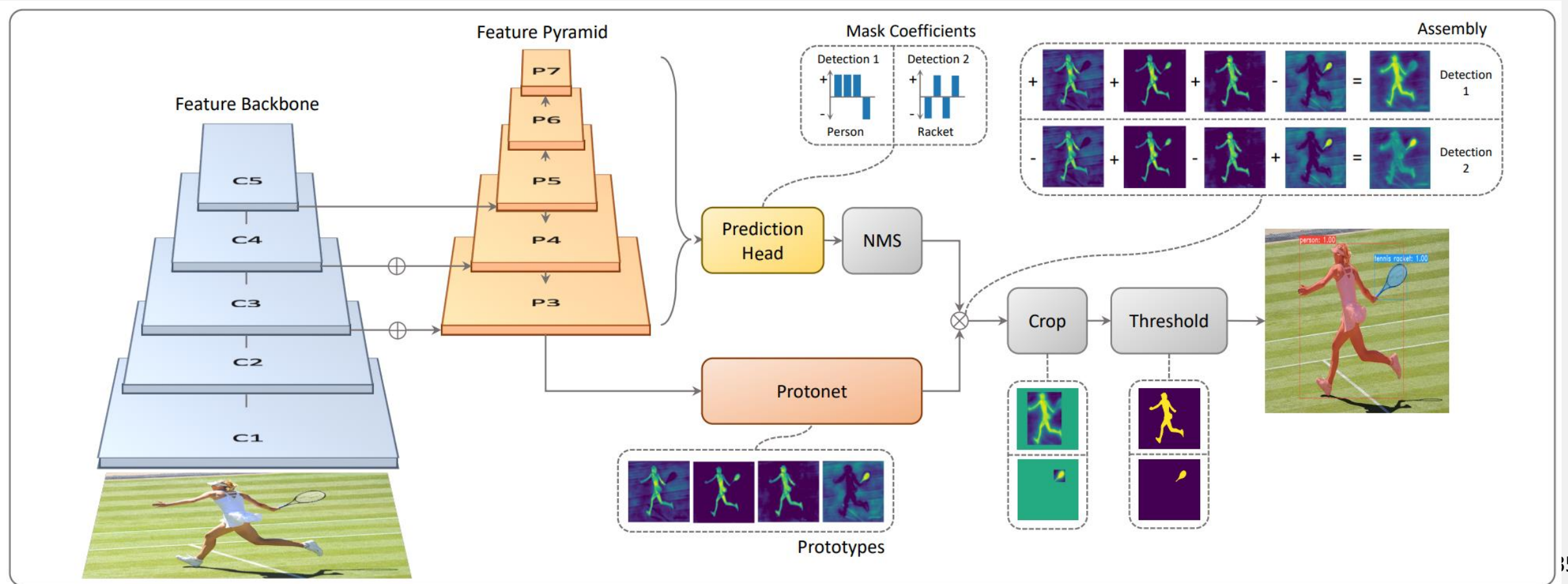
[Mask Scoring R-CNN](#)

# CNN Model

## Real-time Segmentation

➤ YOLACT (You Only Look CoefficientTs)


[Reference](#)



# CNN Model

## Referring Image Segmentation

*...standing*                      *...someone*                      *...bat*

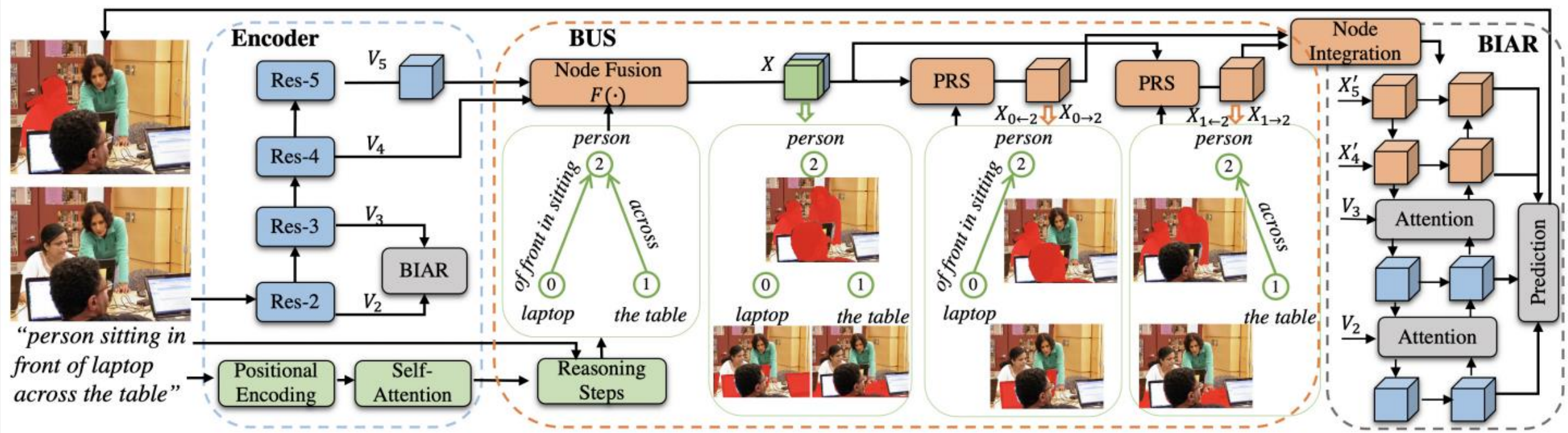


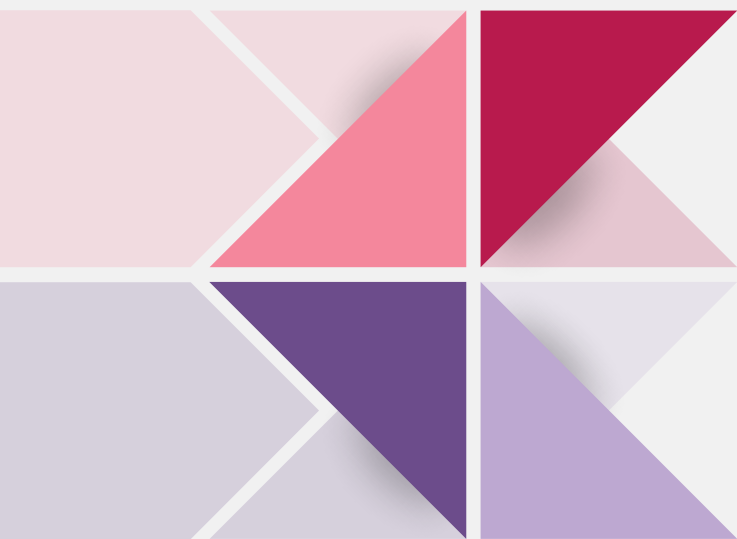
*Man in a vest and blue jeans standing watching someone swing a bat.*

[Reference](#)

# CNN Model

## Bottom-up Shift



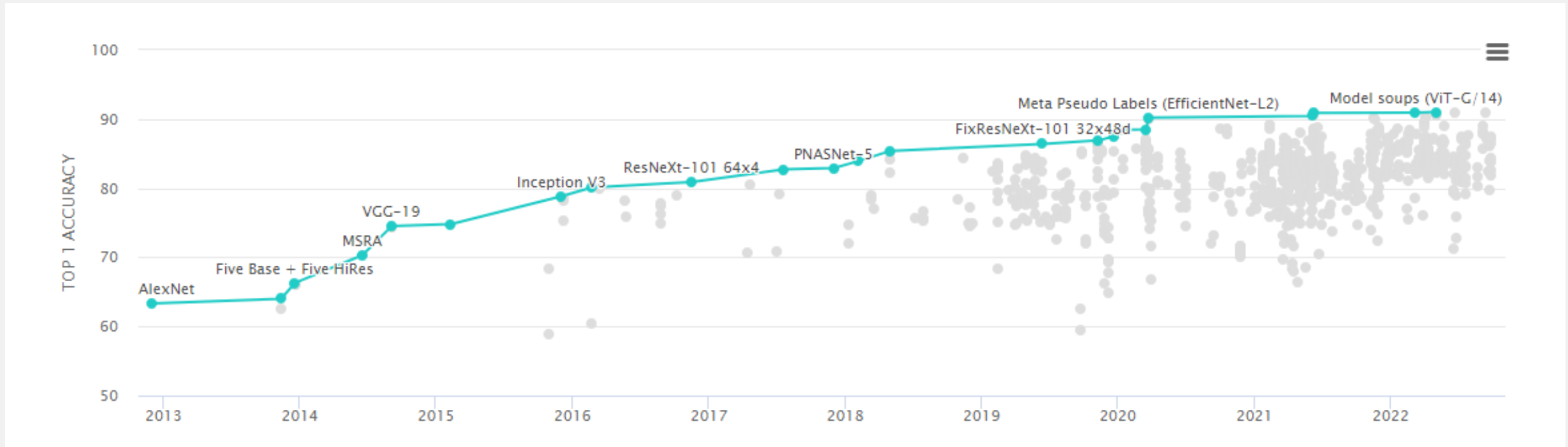


**03**

# **Classification**

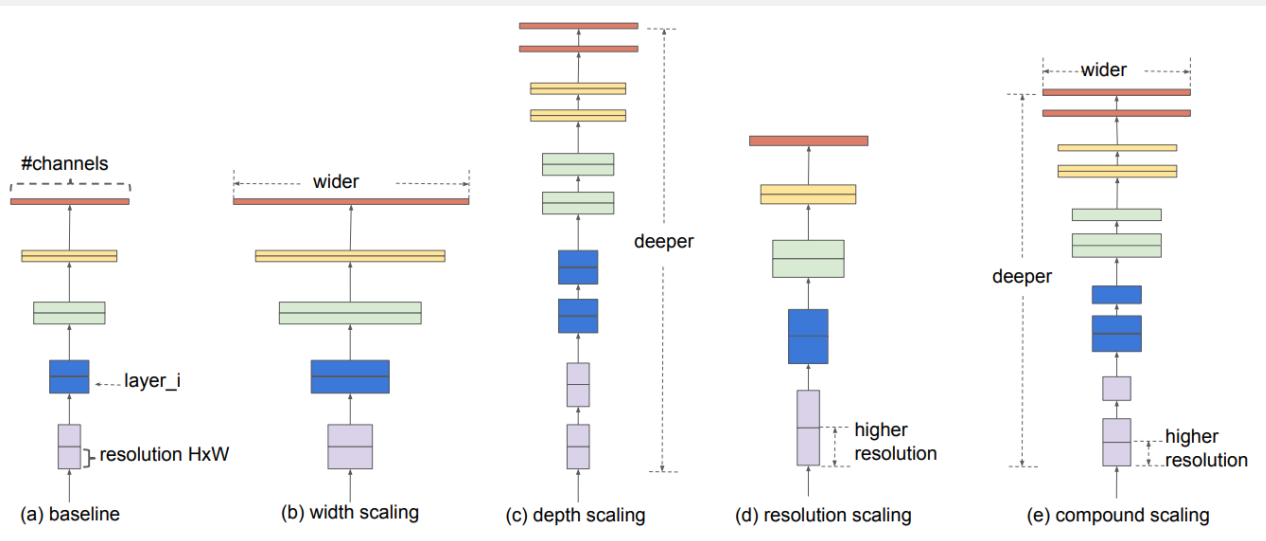
# CNN Model

## Object Classification

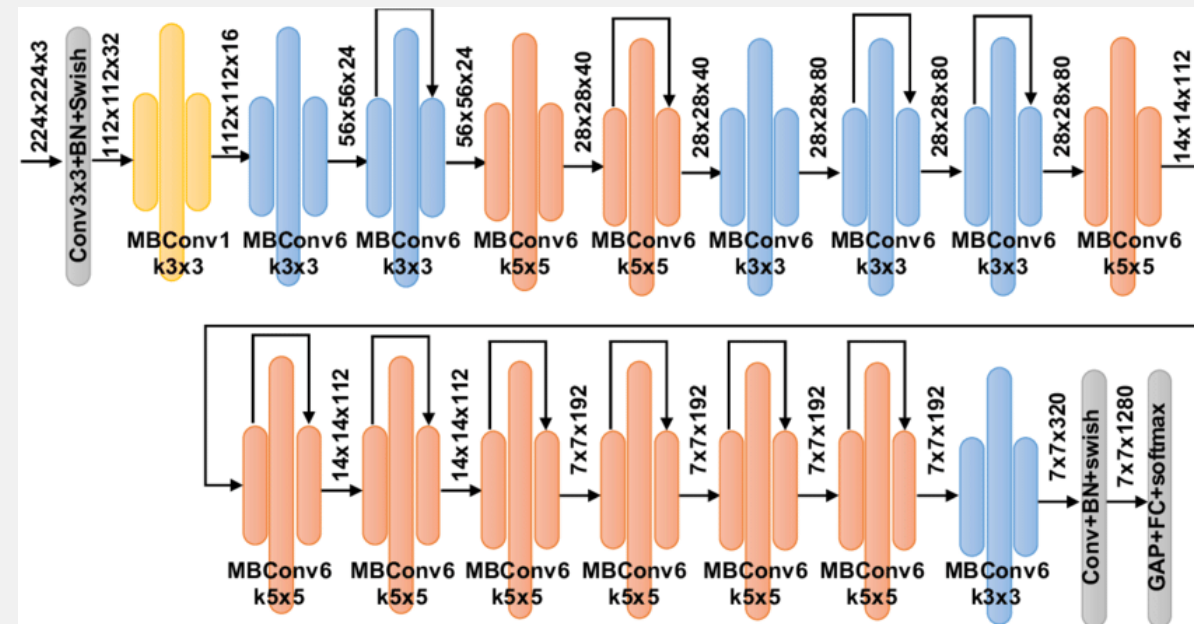


# CNN Model

## EfficientNet



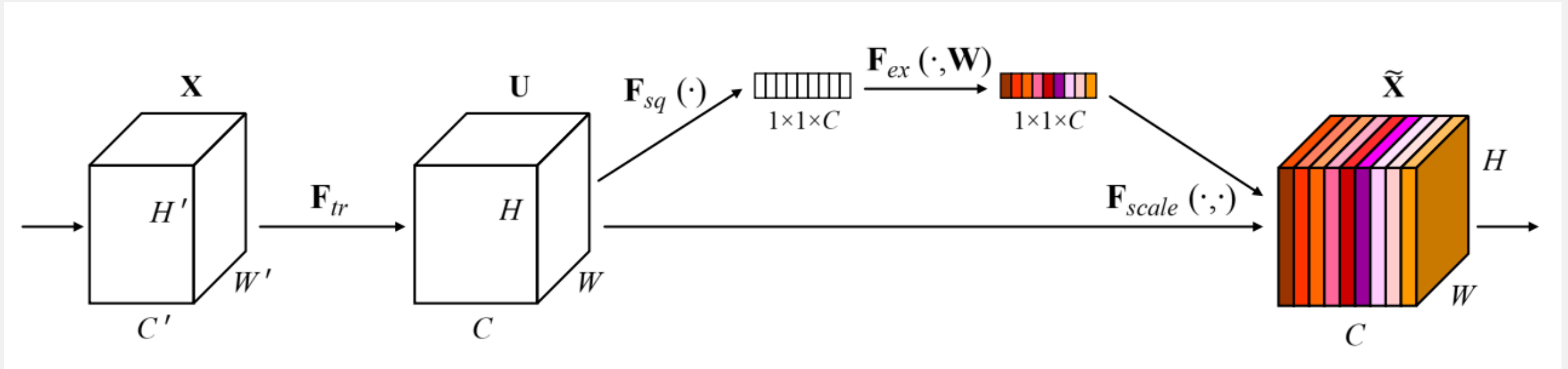
### Model Scaling



### EfficientNet-B0

# CNN Model

## SENet (Squeeze and Excitation Network)

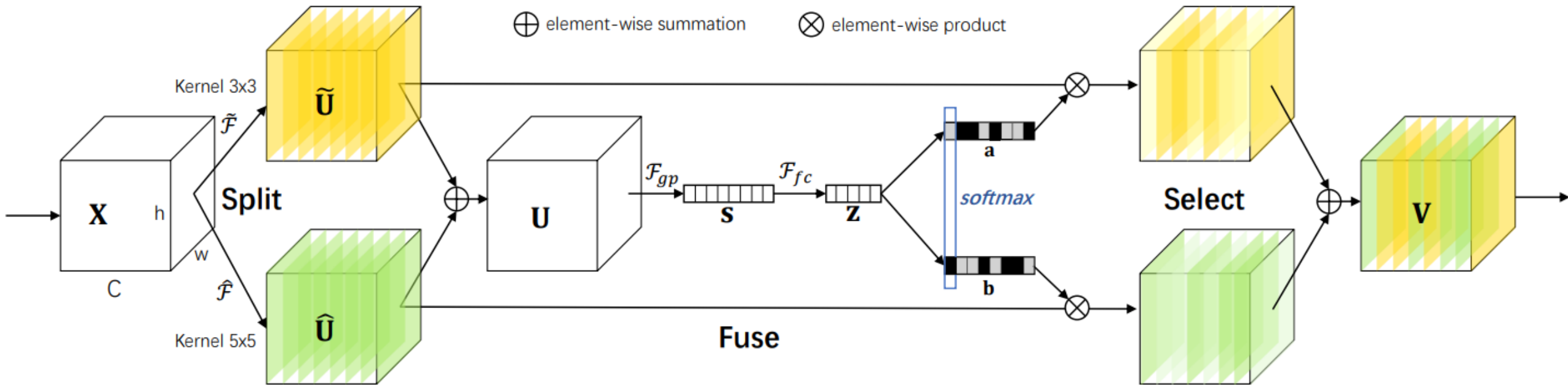


[Reference](#)



# CNN Model

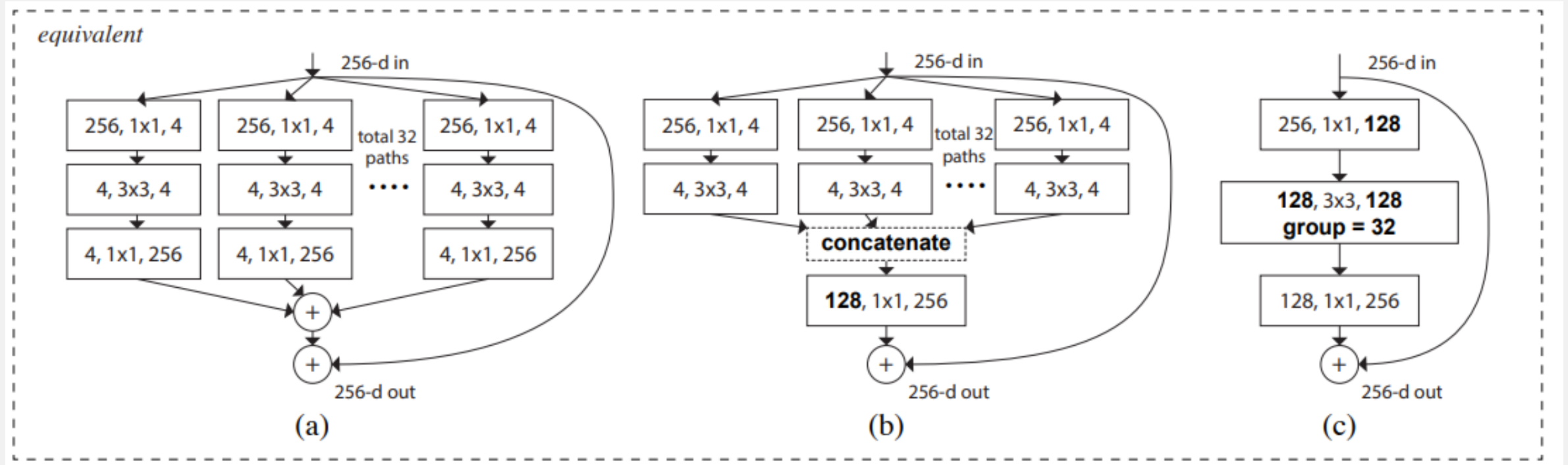
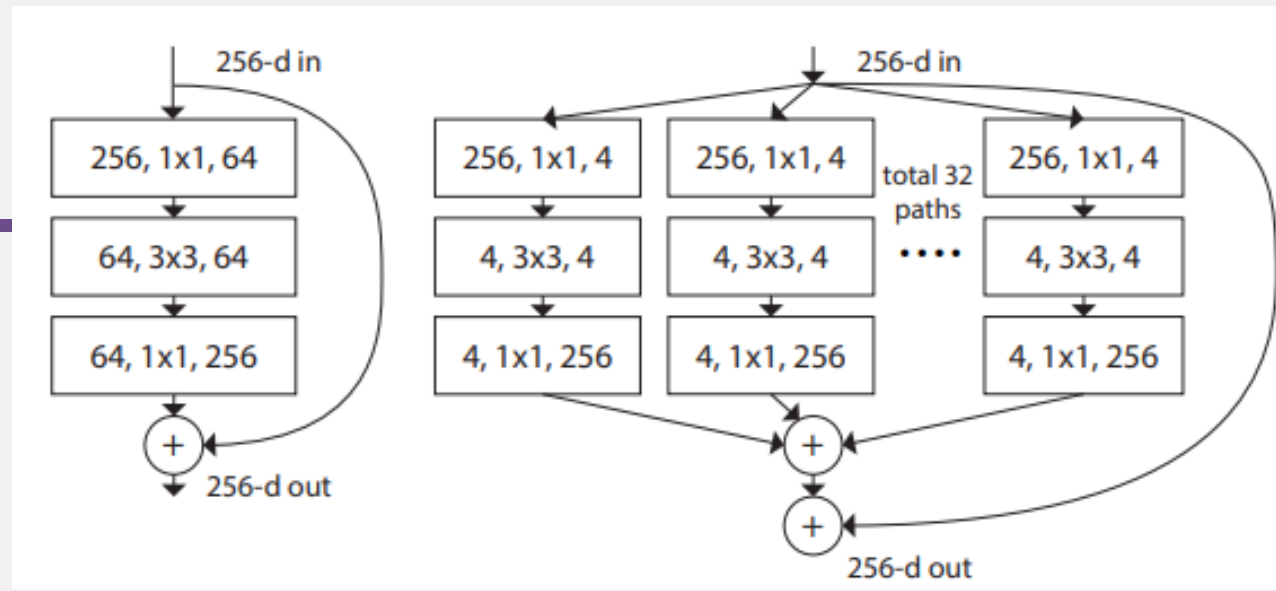
## SKNet (Selective Kernel Network)



[Reference](#)

# CNN Model

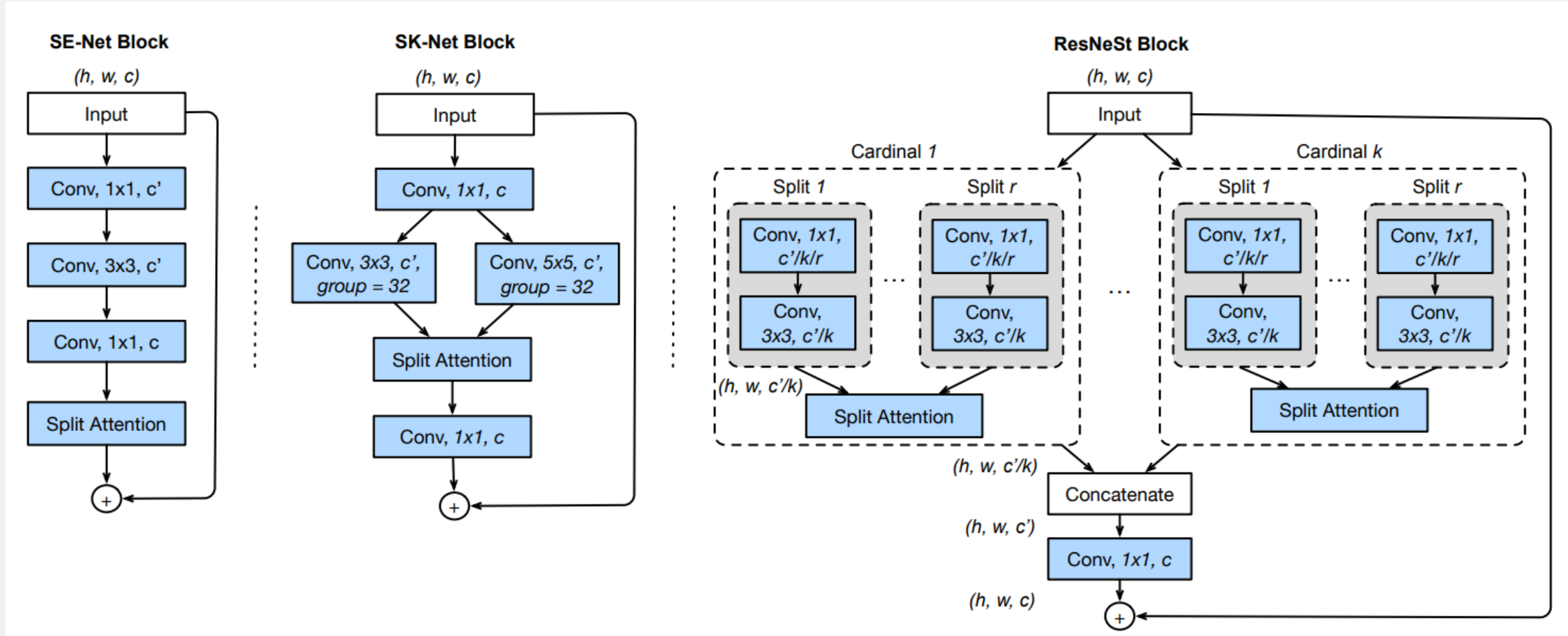
## ResNeXt



[Reference](#)

# CNN Model

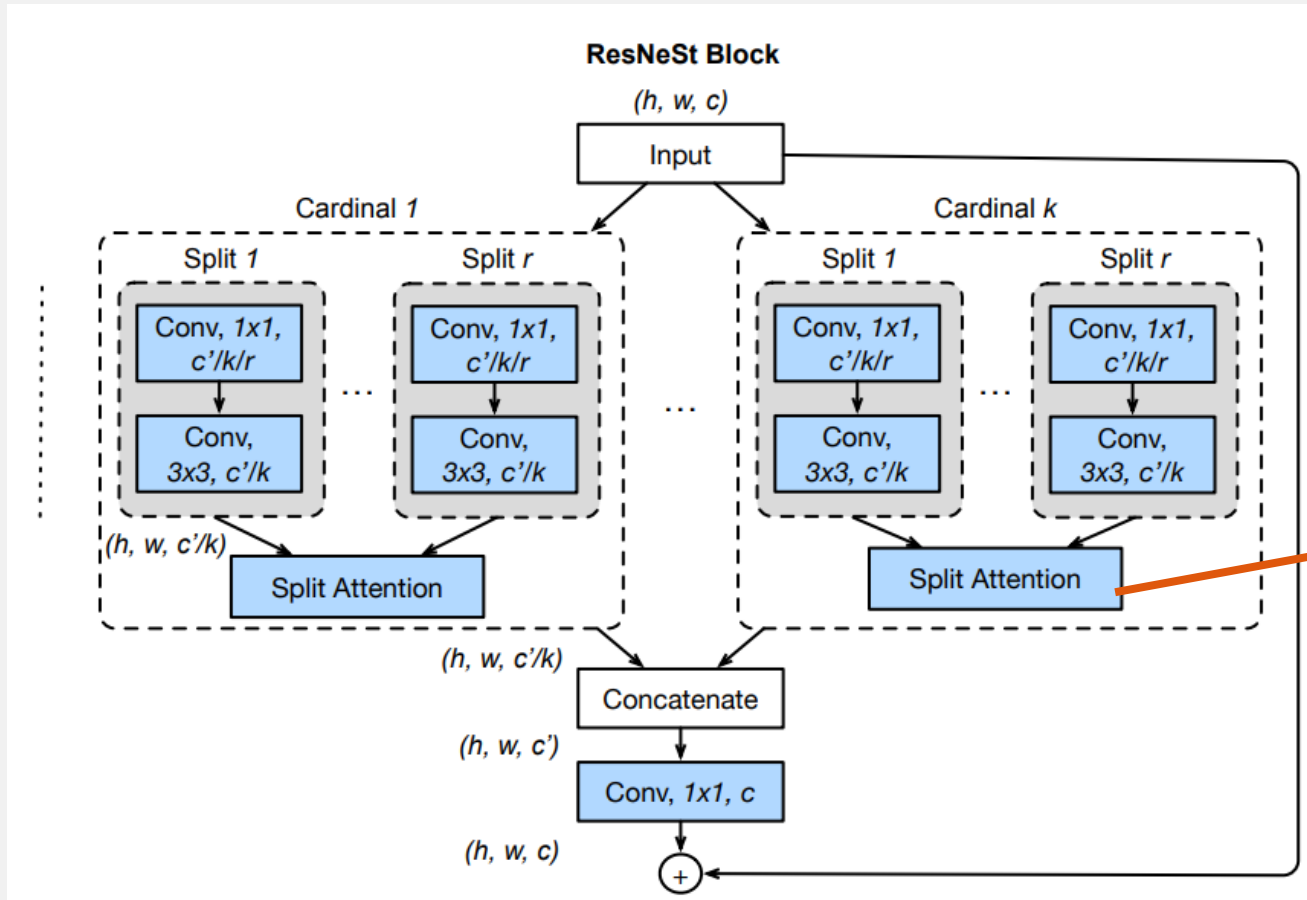
## ResNeSt



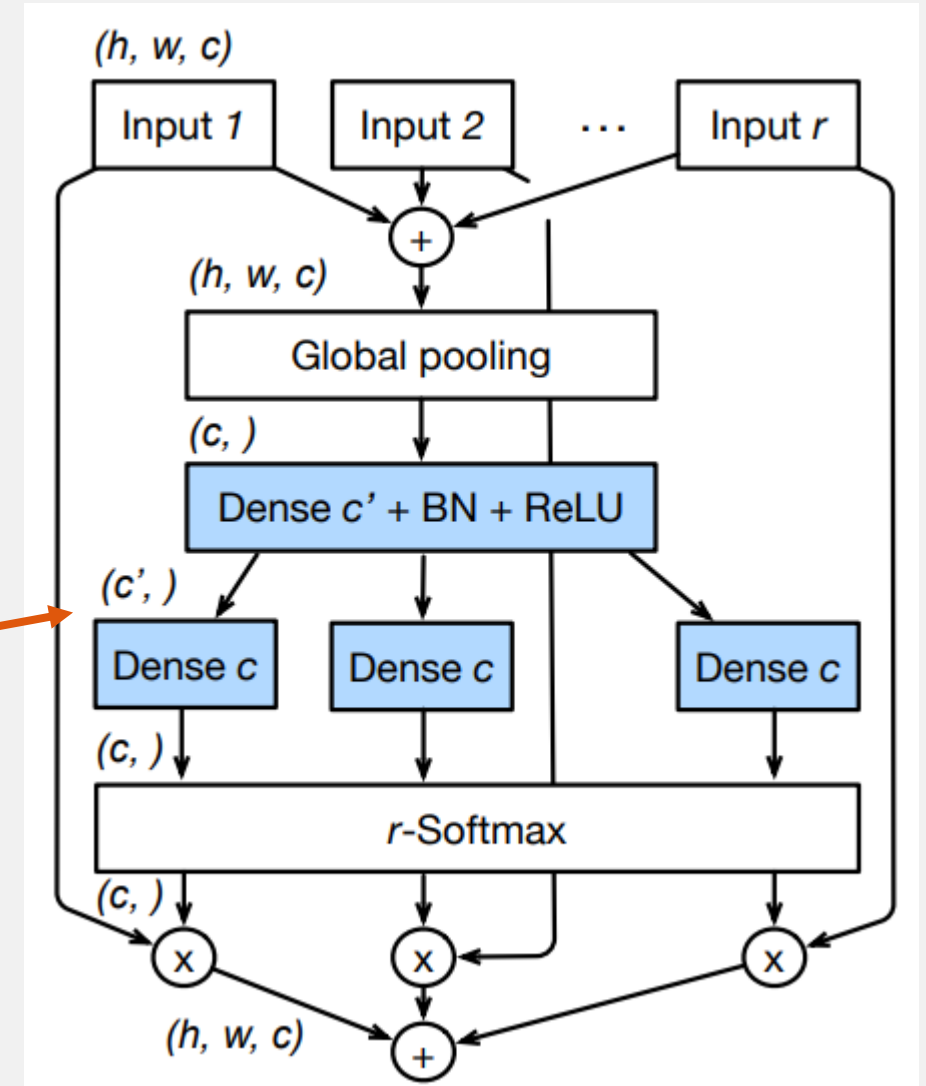
[Reference](#)

# CNN Model

## ResNeSt

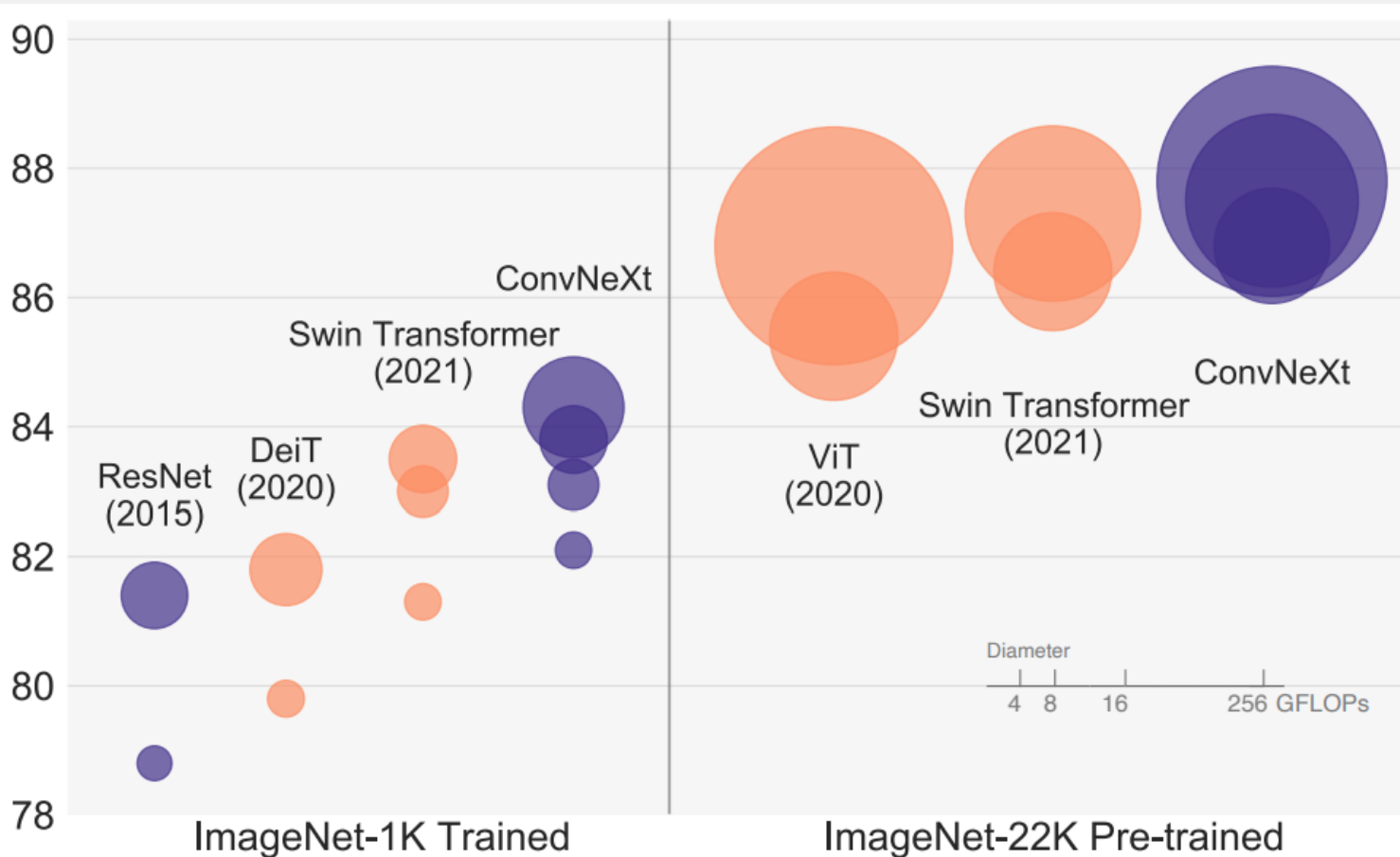


[Reference](#)

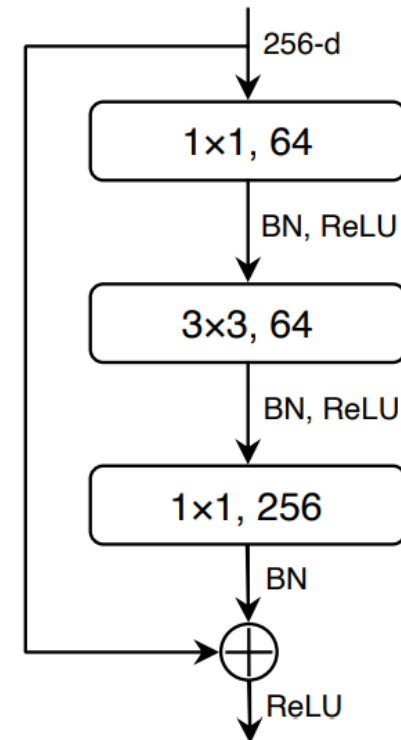


# CNN Model

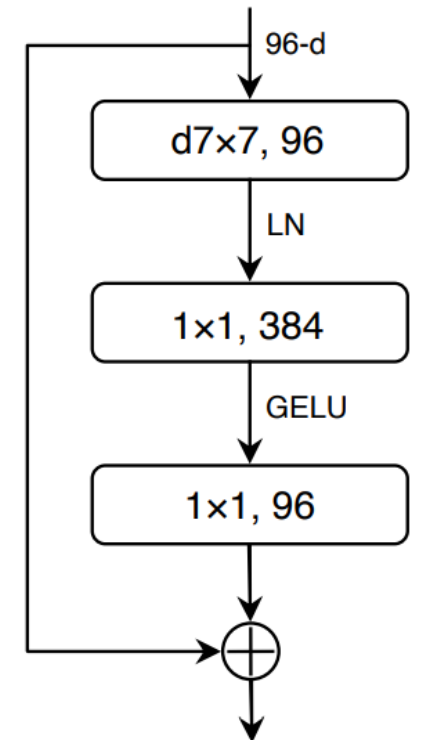
## ConvNeXt



### ResNet Block

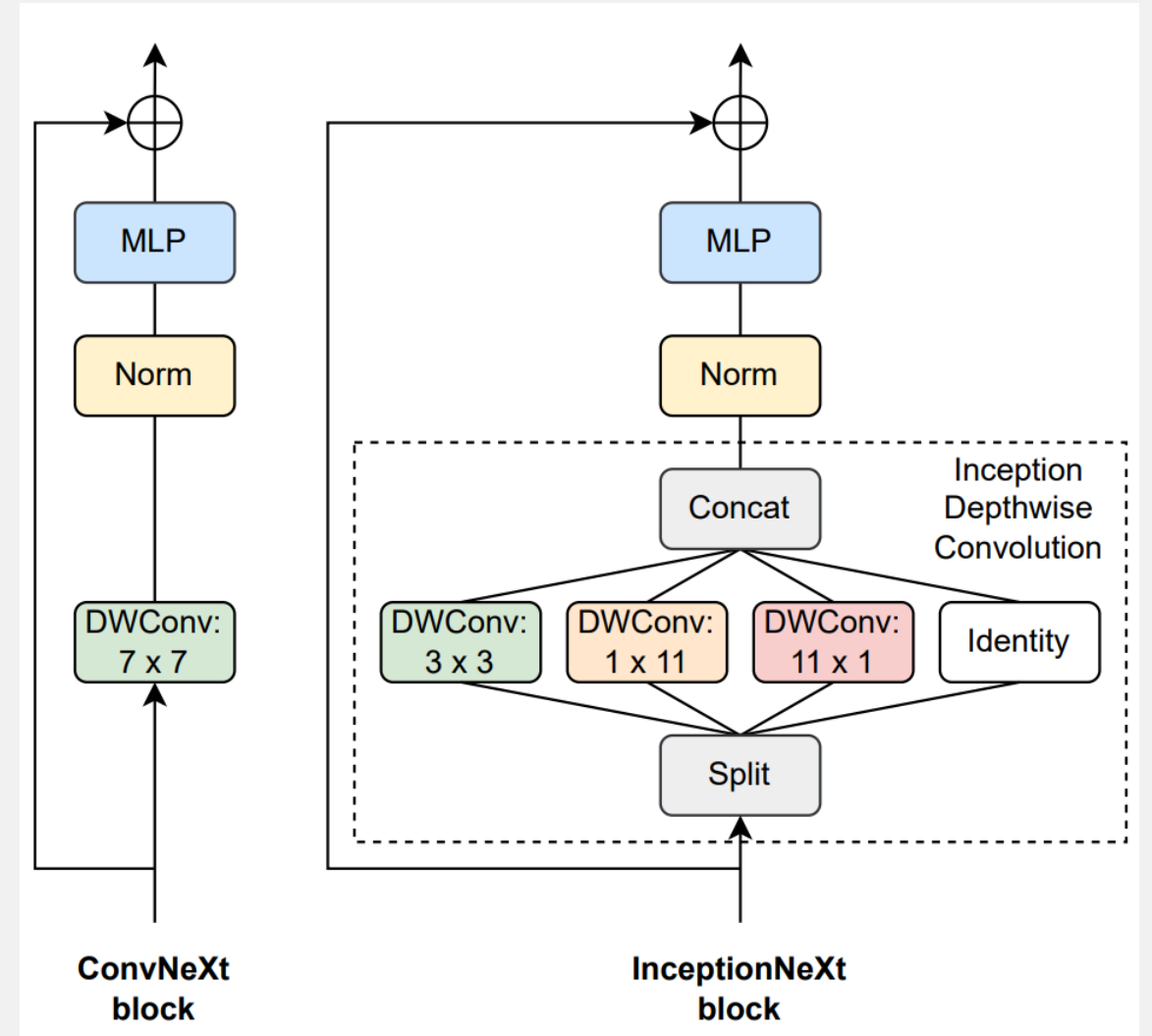
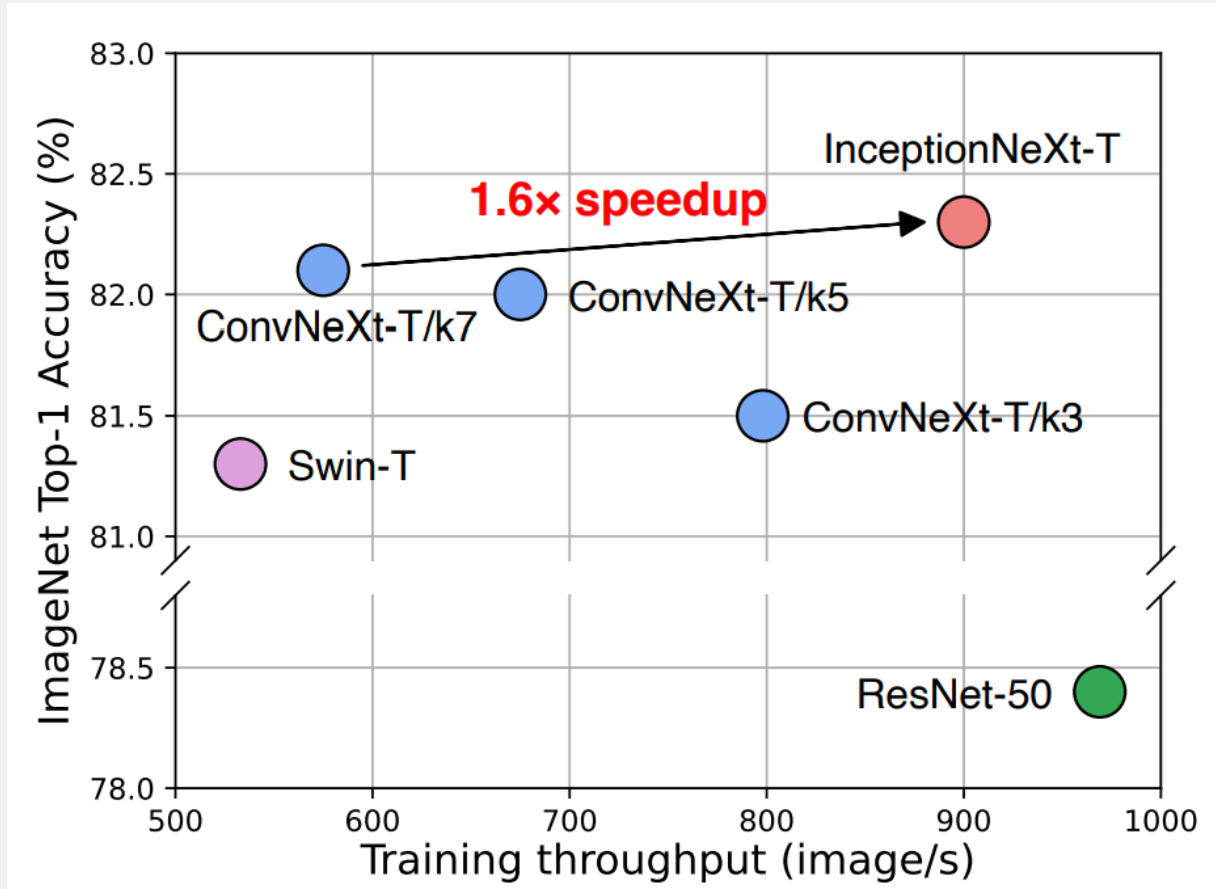


### ConvNeXt Block

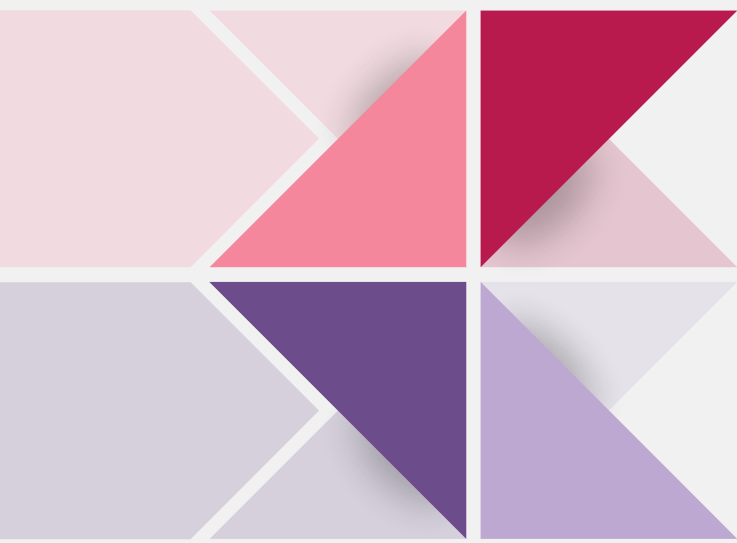


# CNN Model

## InceptionNext



[Reference](#)



**04**

**To date....**

# CNN Model

Module design (or Feature extraction and fusion design)

## Design module rather than architecture creation

- Yolov3 -> Yolov4
- CNN -> ConvNext -> InceptionNext

